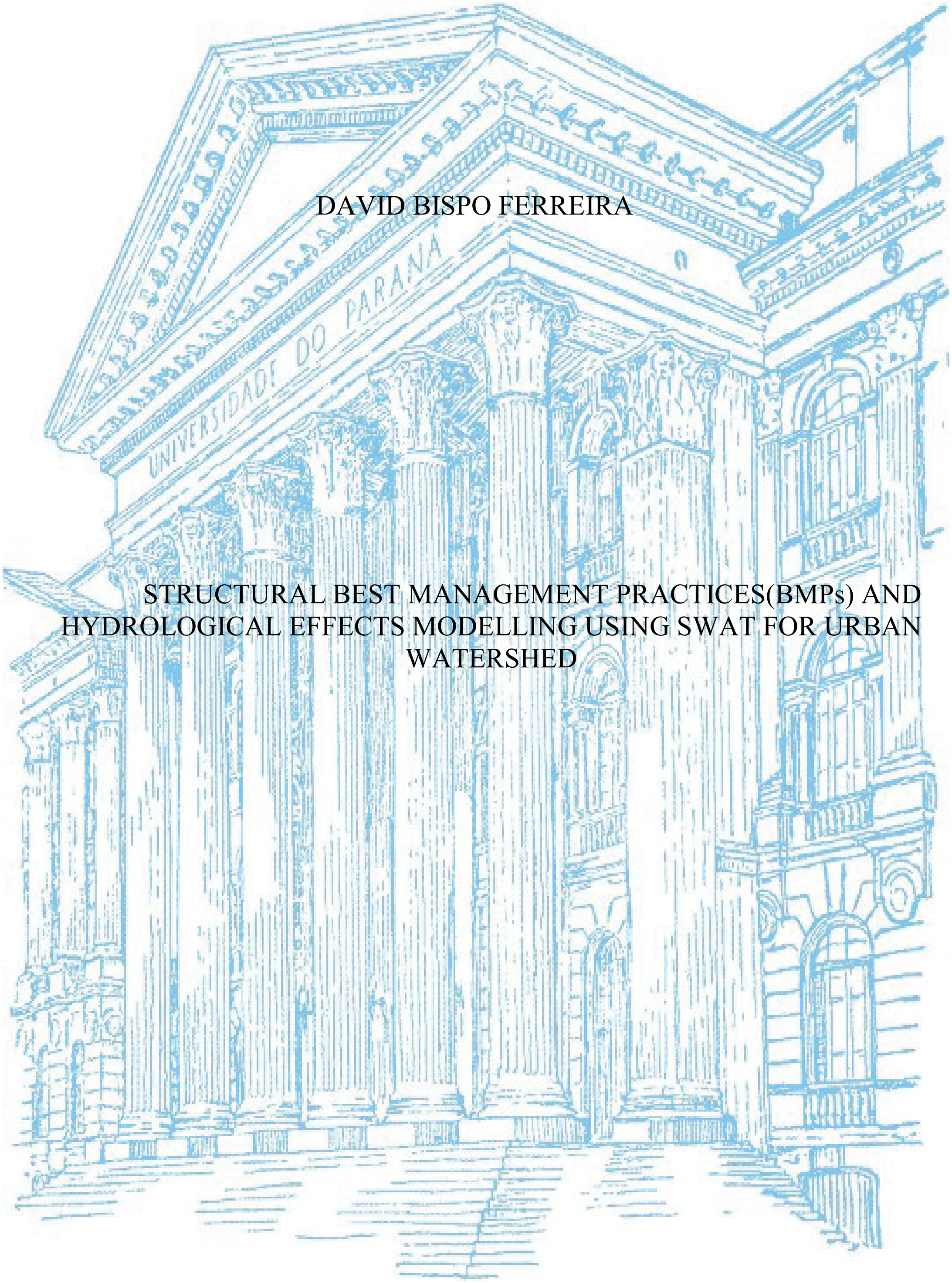
UNIVERSIDADE FEDERAL DO PARANÁ



DAVID BISPO FERREIRA

STRUCTURAL BEST MANAGEMENT PRACTICES(BMPs) AND HYDROLOGICAL EFFECTS MODELLING USING SWAT FOR URBAN WATERSHED

CURITIBA,
2019

UNIVERSIDADE FEDERAL DO PARANÁ

DAVID BISPO FERREIRA

STRUCTURAL BEST MANAGEMENT PRACTICES(BMPs) AND HYDROLOGICAL EFFECTS MODELLING USING SWAT FOR URBAN WATERSHED

Dissertação apresentada ao Programa de Pòs-Graduação em Engenharia de Recursos Hídricos e Ambiental, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial para obtenção do título de mestre em Engenharia de Recursos Hídricos e Ambiental

Orientador(a): Prof. Dr. Cristovao V.S. Fernandes

CURITIBA,
2019

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

F383s Ferreira, David Bispo
Structural best management pratices (BMPS) and hydrological effects modelling using swat for urban watershed [recurso eletrônico] / David Bispo Ferreira. – Curitiba, 2019.

Dissertação - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia de Recursos Hídricos e Ambiental, 2019.

Orientador: Cristovão Vicente Scapulatempo Fernandes .

1. Bacias hidrográficas urbanas. 2. Solos – Análise. 3. Água – Análise. 4. Python (Linguagem de programação de computador). 5. Hidrologia. I. Universidade Federal do Paraná. II. Fernandes , Cristovão Vicente Scapulatempo. III. Título.

CDD: 363.7

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

MINISTÉRIO DA EDUCAÇÃO
SETOR SETOR DE TECNOLOGIA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO ENGENHARIA DE
RECURSOS HÍDRICOS E AMBIENTAL - 40001016021P0

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA DE RECURSOS HÍDRICOS E AMBIENTAL da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **DAVID BISPO FERREIRA**, intitulada: **STRUCTURAL BEST MANAGEMENT PRACTICES (BMPS) AND HYDROLOGICAL EFFECTS MODELLING USING SWAT FOR URBAN WATERSHED**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

A outorga do título de Mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 15 de Março de 2019.

Cristovão V. S. Fernandes

CRISTOVÃO VICENTE SCAPULATEMPO FERNANDES
Presidente da Banca Examinadora

GWYNN ROCHELLE JOHNSSON
Avaliador Externo (PSU)

Nilza M. R. Castro

NILZA MARIA DOS REIS CASTRO
Avaliador Externo (UFRGS)

IRANI DOS SANTOS
Avaliador Externo (UFPR)

DANIEL HENRIQUE MARCO DETZEL
Avaliador Interno (UFPR)

ACKNOWLEDGEMENTS

I would like to thank first of all Cris. Without his support and counselling I would not be writing this document at all. Also, for believing in me and in my ability to conduct this project. This project started in Portland, OR, where I met Dr. Gwynn Johnson, which in addition to teaching me environmental science and American culture, also paved the way for me to become a rather capable (I hope) environmental analyst.

My parents also deserve a big thank you, for giving me a lot of support, creating a pleasant environment, keeping me motivated and making me believe in myself. Gabriela, my muse, my inspiration. Thank you for helping me through the bad and good moments of this work, and to contributing and understanding to the one thing which has took me many hours of our beloved time together.

Also, to the great friends and people who's experienced tailed this work and refined our research, both in the Post-Graduate Program and at the Infrastructure and Transportation Technology Institute. The last deserves an honorable mention, as it also has been a great ride, both personally and professionally. To all the ITTI team, thank you for making my days. I have been learning a bunch and having a blast!

RESUMO

As Best Management Practices (BMPs) têm sido usadas como solução para mitigação de condições de pós-desenvolvimento em bacias urbanas e rurais. Estes dispositivos regulam vazões e volumes, além de capturar poluentes do escoamento superficial usando vários mecanismos. Estes dispositivos têm sido estudados e seu uso disseminado em vários países. Concomitantemente, o melhoramento de modelos de transporte e destinação de constituintes para investigar os efeitos, algoritmos para otimizar a busca por locais ótimos de instalação e facilitação da avaliação de entradas e saídas trouxe à luz vários desafios no que tange a modelagem dos fenômenos, incluindo a seleção de escalas de dimensão e tempo adequadas à representação dos fenômenos. A revisão de literatura demonstra uma fronteira clara entre usar inputs massivos de dados e computação exaustiva em modelos para descrição detalhada dos processos ou a adoção de abordagens mais simplificadas que capturem áreas maiores a custos menores de levantamento de dados. Neste estudo o Soil and Water Assessment Tool (SWAT) é utilizado como solução harmônica para modelagem em bacias com usos do solo mistos. Para vencer os desafios acima citados, BMPs são tratadas como zonas de recarga, isto é, zonas com Números de Curva (CN) menores. A localização destes dispositivos no modelo é realizada utilizando critérios consolidados de viabilidade através de ferramentas já desenvolvidas. Quatro cenários de redução percentual são utilizados para avaliação das melhoras de fluxo nas escalas da Hydrological Response Unit (HRU), subbacia e curso do rio(reach): 10%, 30%, 50% e 70%. As mudanças foram avaliadas na escala diária e anual, usando aplicações desenvolvidas em Python para automatizar a parametrização do modelo e a entrada e saída de dados. O estudo foi bem-sucedido em conceber a geração de múltiplos cenários, assim como em produzir ferramentas que auxiliem a entrada e saída de dados. Os resultados demonstram que a criação de zonas de recarga é mais eficaz em regiões onde há mais capacidade de retenção do solo. Do contrário, a redução do escoamento superficial tende a chegar em um limite, a partir do qual não há mais roteamento do escoamento superficial. Em HRUs e subbacias onde as condições de solo são favoráveis, a dinâmica de roteamento superficial e subsuperficial é modificada, fazendo com que a recarga dos aquíferos aumente e as recessões sejam mais lentas. Em geral, não são visíveis efeitos na escala da subbacia e no curso principal do rio, uma vez que muito do escoamento superficial é roteado como escoamento lateral ou fluo de subsuperfície. Além disso, a superposição dos efeitos para o resto da bacia é muito pequena na escala diária.

Palavras-chave: SWAT. Bacias Urbanas. Python. Best Management Practices Hidrologia.

ABSTRACT

Best Management Practice (BMP) devices have been employed as a solution for both agricultural and urban watershed post-development effect mitigation. These devices regulate flow and capture runoff pollutants using various mechanisms. Such devices have been studied and its use disseminated in several countries. Concurrently, the enhancement of pollutant fate and transport models to assess the effects, search for optimal locations and facilitate inputs has brought to light several challenges concerning the modelling of physical phenomena, especially the one related to selecting time and size scales for adequate representation. The literature revision demonstrates that a clear boundary between using massive data inputs and computation-exhaustive models for thorough process description or more simplified approaches that capture larger areas at a more affordable data cost has limited the comprehension and description of BMP hydrological processes at the subbasin and watershed scale. In this study, SWAT is used a harmonic solution for modelling mixed land-use watersheds. To overcome the challenges stated, BMPs are treated as recharge - lower Curve Number (CN) zones, in feasible scenarios generated using an pre-built-tool and consolidated feasibility topographic, hydrological and space-distribution features. Four scenarios were generated: 10, 30, 50 70% CN reductions were tested and evaluated at the daily HRU/subbasin and subbasin yearly average scales, using developed applications for automating the parameter change and Input/output operations. The study was successful in automating the BMP scenario generation and multiple scenario generation as well as output data analysis. Results show that the creation of recharge zones is more effective at regions where more soil storage is available. Otherwise, runoff reduction tends to reach a limit. In HRUs and subbasins where soil conditions are favorable, the entire soil water and groundwater flow dynamics is modified, causing aquifer recharge to increase on average and recessions to be slower. Generally, no effects can be noticed at the subbasin o reach scale, as much of the runoff is also routed either as lateral flow or groundwater flow. The superposition of such effects to the rest of the watershed results in small differences at the daily scale.

Keywords: SWAT. Urban watersheds. Python. Best Management Practices. Hydrology.

SUMÁRIO

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AAT – All at a time

ANA – Agencia Nacional de Águas (National Water Agency)

ARM – Agricultural Runoff Method

AGNPS – Agricultural Non-Point Source Pollution Model

ANNGNPS – Annual Agricultural Nonpoint Source Pollution Model

AT – Almirante Tamandare Flow monitoring station

BMP – Best Management Practice

CN – Curve Number

CONAMA – Conselho Nacional do Meio Ambiente (National Environmental Council)

COALIAR - Comitê das Bacias do Alto Iguaçu e Afluentes do Alto Ribeira

CSA – Critical Source Areas

CX – Ponte da Caximba Flow Monitoring Station

DEM – Digital Elevation Model

DOC – Dissolved Organic Carbon

EMATER - Instituto Paranaense de Assistência Técnica e Extensão Rural

FCCP – Fuzzy-credibility Constrained programing

GA - Genetic Algorithm

GI – Green Infrastructure

GAML – Green-Ampt-Mein-Larson

GIS – Geographical Information Systems

HSA – Hydrological Source Areas

HSP-F – Hydrological Simulation Program – Fortran

HRU – Hydrological Response Unit

LULC – Land Use and Land Cover

LiDar - Light Detection And Ranging

LID – Low Impact Development

NSGA – Non-sorted Genetic Algorithm

NSE – Nash-Sutcliffe Efficiency

OAT – One-at-a-time

OR – Oregon, State in the US

SedFil – Sedimentation-Filtration Basin

PARASOL – Parameter Solution

pH – Hydrogen ionic Potential

PSO – Particle Swarm Optimization

PLOAD – Pollution Load Application

PNPI – Potential Nonpoint Pollution Index

RMSE – Root Mean Square Error

SuDS – Sustainable Drainage Systems

SCIMAP - Sensitive Catchment Integrated Modelling Analysis Platform

SUFI2 – Sequential Uncertainty Fitting Procedure v2

SEMA – Secretaria de Estado de Meio Ambiente (Environmental State Agency)

SQ – Santa Quiteria Flow Monitoring Station

SSS – Storm Separate System

SWAT – Soil and Water Assessment Tool

SWMM – Stormwater Management Model

SWAT-CUP – SWAT Calibration and Uncertainty Program

TAIOM - Topography Analysis Incorporated Optimization Method

TSS – Total Suspended Solids

TR – Technical Release

TX – Texas, state of the US

USLE – Universal Soil Loss Equation

USEPA – United Stated Environmental Protection Agency

USDA – United States Department of Agriculture

WWTP – Wastewater treatment plant

TERMS AND PARAMETERS GLOSSARY LIST – SWAT

ALPHA_BF – Baseflow recession coefficient

ALPHA_BNK - Reach flow recession coefficient

CH_K1 – Main channel Effective saturated hydraulic conductivity

CH_K2 – Tributary Channel Effective saturated hydraulic conductivity

DA_ST – Deep Aquifer Storage

DEEPST – Initial deep aquifer storage

GW_DELAY – Groundwater flow lag coefficient

GWQMN – Threshold depth of water in the swallow aquifer required for return flow to occur

GWQ – HRU Groundwater flow

LAT_TTIME – Lateral Flow lag coefficient

RCHRG_DP – Partitioning coefficient for distributing percolated water between shallow and deep aquifer

SOL_K – Soil Saturated Hydraulic conductivity

SHALLST – Initial shallow aquifer storage

SURLAG – Surface Runoff lag coefficient

SA_ST – Shallow Aquifer Storage

SLSOIL – Slope Length for lateral subsurface flow

SLSUBBSN – Average Slope Length

SURQ – HRU Surface Runoff

SW – HRU Soil Water

TXTInOut – Text Folder with the SWAT runtime input documentation

WYLD – HRU/Subbasin Water Yield (GWQ+SURQ+LATQ)

FOREWORD

This project was born in Portland, OR, home to this author for 10 months. Dr. Gwynn Johnson and I thought of a simplified index for quantifying potential pollution using GIS systems and multicriteria decision. With a lot of lessons learned, one question remained: “how much do these devices really work on a from a larger perspective? Perhaps the watershed scale?” It just so happens that more researchers have found themselves asking the same question. And as a paradox between too much data for a detailed model and too much approximations for simplified models developed, we propose to adapt SWAT to assess the hydrological and water quality effects in urban environments in a watershed scale.

This project contemplates the assessment of hydrological effects from BMPs (please keep in mind that this study conceptually simplified BMPS as recharge zones) through appending BMP features to Land Use Maps and reparametrizing them. Originally, it was intended to assess water quality improvements, which is a step further to hydrological calibration and analysis. This work has concentrated in showing in various scales, what are the expected watershed benefits, of implementing BMPs, whether the HRU, which in this case is the in-situ scale, subbasin or watershed.

This work will be continued, as the authors are engaged in finding the answer to those questions and see potential for contributing to the subject literature.

1 INTRODUCTION

Development and land use changes/land uncover causes impacts in watersheds hydrological responses, most especially watersheds with urban characteristics. The reduction of permeable areas in urban centers causes faster responses to rainfall events in watersheds, higher flow peaks and shorter travel times. Along with hydrological issues, stormwater may also contain substances originated from diffuse pollution, which can reach concentrations similar to raw sewage for many water quality parameters. The degradation of aquatic environments by diffuse pollutants may endanger human safety, since water resources are employed for human consumption and also for recreational purposes (USEPA ,1999).

Low Impact Development (LID) and Best Management Practices (BMPs) consist on a series of engineering practices to mitigate development impacts on watersheds. They include but are not limited to structural solutions, such as (i) grassy swales, (ii) vegetated roofs and (iii) planted wetlands to control flow volume and enhance water quality. The performance of these devices is well documented in the literature and has been successfully implemented around the world. These devices are also an important tool to achieve water quality standards under various legal frameworks (ASCE, 2015; KELLAGHER et al., 2015).

The main approach to understand the problem is to quantify and to assess water quality parameters as consequence of hydrological influences. Their values help to understand the environmental speciation at a time in space. Most times, these procedures are performed through sample collection, like grab sampling or automatic samplers. Other techniques, like Lidar Scanners and Remote sensing, facilitate the task of building a discrete film, capturing discrete time variations. Diffuse pollution modelling is a complimentary tool, that has been employed for several years in an attempt to spatially account for point and nonpoint sources. It assists water resource managers to comprehend diffuse pollution causes and create mitigation strategies (YANG; WANG, 2010).

In such a context, the application of diffuse pollution models has gained strength with GIS software development, as programmed routines allow Geoprocessing platforms to compute spatial parameters. GIS applications provide Graphical User Interfaces (GUI) and populate input data tables for model run (BORAH; BERA, 2004). Within the distributed pollution models, SWAT is a powerful tool employed in several locations. It is a distributed, multi-scale model, that performs water balances in daily and sub daily timesteps. SWAT considers a land phase and channel phase, also accounting for groundwater recharge and transport. Routines for hydraulic and water quality management structures are available in the model (NEITSCH et al., 2011).

This study aims to assess the efficiency of implementation three BMP devices in the Barigui river basin using SWAT, through coupling GIS programming for potential BMP site identification and SWAT data generation processes. For scenario exploration and data analysis, Python data-science tools were employed to explore parameter variation and data analysis at the HRU and subbasin scales. This work expects to contribute to this existing gap between field-scale BMP effects and their effects on the watershed level.

In this research, The Barigui Basin is an urban basin with very irregular land use cover. And the focus of the case study. Frequently, the water quality standards of the Barigui River are below thresholds set by the National Environmental Council (*Conselho nacional do meio ambiente* – CONAMA) (BRASIL, 2005). (Goncalves; Fernandes, 2008; SEMA, 2014; Kozak, Fernandes, 2016). Flow and water quality monitoring has been conducted in the Basin by both the Federal University of Parana and public institutions. Hence, a diffuse pollution model could point major sources of diffuse pollution and where the flow and diffuse pollution abatement would be more significant using BMP devices.

1.1. Evidences of interest

Urban Land occupation in Brazil has provoked significant changes in watersheds hydrological cycles. At the time, the employed solutions comprehended a “storm-drain to river” approach. However, these solutions incurred in flooding problems as more runoff was transported as non-permeable areas were created. Detention tanks have been proposed in many cities in Brazil as ways to mitigate flood impacts on the watershed. However, such solutions do not seek reduction on water quality impacts (CANHOLI, 2015).

Several studies showed that diffuse pollution is an issue that can compromise surface water quality (SHAVER et al., 1994 ; Burton; Pitt, 2001 and Novotny, 2003) . The combination of diffuse pollution models with sampling efforts has been widely used to better understand the involved mechanisms and draw management scenarios. Figure 1 shows a conceptual model for watershed modelling (NOVOTNY, 2003).

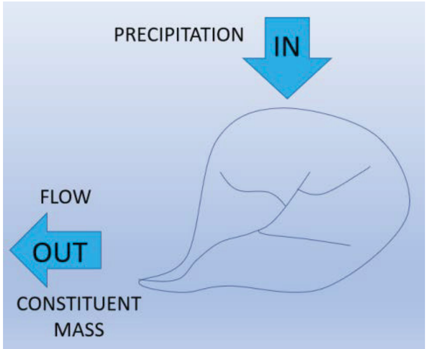


Figure 1– Conceptual model for watershed modelling

Water quality in Urban Watershed , according to previous studies, , reports that the water quality is, in general in uncompliant with the water regulations (BRASIL, 2005, COALIAR, 2013).. Thus, is requires the existence of high-resolution GIS data and length sufficient rainfall gaging, and high frequency flow gaging stations suggests potential for efficient hydrological and water quality distributed simulation to study the diffuse pollution problem in these applications. The Soil and Water Assessment Tool (SWAT) has been successfully employed in cases of low data availability providing good results in assessing BMP effects, although sub daily and BMP computation routines are still in development (HER et al., 2017 ; JEONG et al., 2010, 2013). BMP ability to mitigate flood and water quality development impacts has been extensively demonstrated. It reduces flood peaks and volumes, as infiltration and detention are fundamental principles of such devices. Streambank erosion and water quality is significantly improved solids, and adhered constituents are also removed (DIETZ, 2007). The need to comprehend the mechanisms related to diffuse pollution and BMP employment and its consequences to water quality is important for water management, and according to the literature, not completely understood, as is difficult to expand time-sensitive field-scale hydrology to the totality of the watershed (HUNT et al., 2009;AHIABLAME et al., 2012 ; ELLIOTT, TROWSDALE, 2007).

This work explores the creation of Python applications for the generation of 3 BMPs scenarios for SWAT evaluation. BMPs are simplified to recharge zones, or zones of reduced Curve Number. To complement the strategy and make feasible to explore different parameters, other Python Input/output tools were developed. The goal is to evaluate at the HRU and subbasin daily scales. These results then are expanded to the year average scales, in order to evaluate the mass transfers typically taking place the watershed.

1.2. How this document is structured

The contents on this document are structured according to the following division:

(i) Introduction, where diffuse pollution, causes and effects are introduced. This section also contains the hypothesis and objectives of this work.

(ii) Literature revision: A brief review of the main topics that pertain to this work.

In this section, guidelines are to describe conceptual and computational aspects, in a time-organized fashion, the development of models and features to this day, their advantages, and main differences in terms of water quality and quantity. At first, presents what are BMPs, its known advantages, mechanisms, performance evaluation, and design guidelines. Then, the concepts of diffuse pollution modelling, its aspects, performance conditions, data source requirements and main model types are presented. Next, state-of-the-art advances on the pollutant fate and transport models resulting from pre-screening as potential solutions to the Barigui River basin problem. Finally, the Soil and Water Assessment tool is presented, along with its current advances and future directions for BMP modelling.

(iii) Materials and Methods: The methodology of the study is presented. Next, a hypothesis and simplifications section. Data sources and calibration procedures are presented in sequence. Finally, the description of the procedure to modify SWAT inputs, and the programs utilized for such.

(iv) Results: This section brings results, divided by: (a)calibration, (b)final BMP layouts and (c)hydrological results. Each section presents results obtained and selects topics to be further discussed in the final remarks section

(v) Final Remarks: An integrated analysis of this work is presented, through analyzing in a synergic fashion the hypothesis, methods and results, their advantages, limitations, and what are the future directions of this research.

2 OBJECTIVES

2.1 General objectives

This work aims to investigate the nature of the watershed benefits achieved through implementing recharge zones on watersheds

2.2 Specific Objectives

- Simulate adequately hydrological features and behavior of the watershed with the available data.

- Develop tools for scenario generation and Input/output analysis to enable the multiple scenario evaluation

- Initially tackle the evaluation of HRU and subbasin hydrographs at various timescales in the Barigui Basin, study case of this work.

3 WATERSHED MODELLING – BACKGROUND

A main aspect of human contemporary development is the land conversion for economically viable goals. Conversion to either agricultural (e.g.: poultry farming, agriculture, aquaculture) or urban (e.g.: parking lots, industries, residential areas) and the necessity to manage natural responses from water courses are very important to achieve a sustainable relationship with human development effects.

Various hydrological techniques have been created to assess and predict water resources control, aiming to control and rationally manage resources. Some of the main issues that hydrology seeks to solve is the control of flow, avoiding property damage and potentially health hazards. The control of flooding and the associated risks are a major concern for decision-makers and the general public, in addition to others concerned with the availability or excess of water on a specific waterbody (FANG, 2016).

Concurrently, environmental pollution is another direct consequence of natural and anthropogenic activity. More specifically, the pollution resultant from human activity is directly related to: (i) economic aspects, as the growth of agriculture, industry, transportation and services boosts the consumption of natural resources and causes impacts to ecosystems, (ii) social aspects, once living standards, popular culture and social organization drive economical aspects and resource consumption.

Water pollution can be classified according to the spatial distribution of inputs: point and diffuse pollution (or nonpoint pollution - they have exact meanings). More than one definition has been created to qualify exactly nonpoint pollution, mainly varying according to the framework that the term is built in (U.S. EPA,1983; NOVOTNY, 2003; ESLAMIAN ,2016). One definition is: “pollution derived from land use activities (urban and rural), disperse along a watershed or a sub-watershed, and not sourced from a discrete point, such as industrial effluent from sewage, mining operations and agricultural or animal farming activities.” (FERRIER ,2005). Figure 2 shows a conceptual scheme of surface and groundwater diffuse pollution.

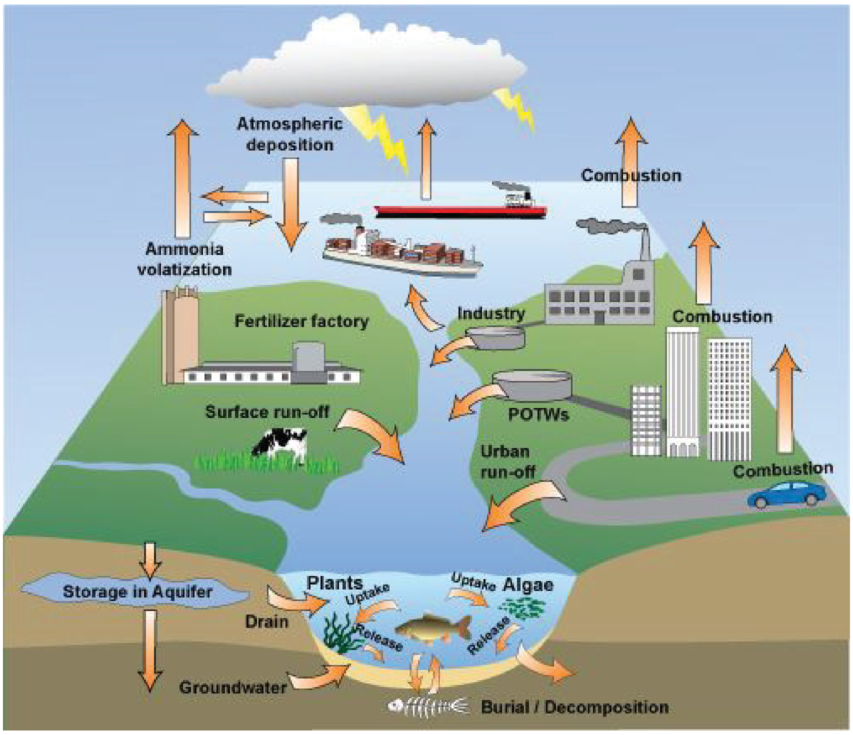


Figure 2 - Diffuse pollution in the watershed

Source: https://www.pueblo.us/1605/Nonpoint-Source-Pollution

Diffuse pollutants include but are not limited to nutrients, hydrocarbonates, heavy metals, solids, pesticides, pathogens, personal care products and other synthetic chemicals derived from industrial activities (USEPA, 1999). Diffuse pollution is directly related with the changes in watershed response to precipitation events, and on stormwater speciation as it scours various surfaces, transporting substances to receiving waterbodies. Changes like the removal of natural vegetal coverage, whether for agricultural and livestock breeding or for urban uses intensify diffuse pollution processes.

Industrial and agricultural activities also contribute for diffuse pollution, as well as the removal of riparian areas and the increase in atmospheric humid and dry deposition (NOVOTNY, 2003). The contaminants present in atmospheric deposition are washed off from the surfaces during rainfall events. Therefore, stormwater quality is highly dependent on atmospheric conditions, quantities of particulate matter and on the constituents, present on the scoured surface. Other source of diffusion pollution is the illegal discharge of raw sewage into rivers. Developing countries are the most susceptible to this kind of diffuse pollution, once they often lack proper sanitation measures (NOVOTNY, 2003).

These changes modify the watershed ability to facilitate sediment replenishment and protect water quality and aquatic health by removing excess nutrients and chemical contaminants before runoff enters receiving waters is drastically reduced (FIELD; TARURI ,2005). Negative aesthetic effects, loss of fish and other sensitive aquatic species, increased temperatures, and increases in sediment transport are a few consequences of diffuse pollution inputs (EPA, 1999; NOVOTNY, 2003).

The challenge in identifying and mitigating diffuse pollution relies on the disperse spatial distribution and pathways, and on the different chemical forms that pollutants may be transported, dissolved, suspended or adsorbed. Furthermore, chemicals are subject to transformations and reactions that may occur as a function of stormwater speciation, pH, redox potential and other various conditions.

3.1 Water and constituent responses in watersheds

Hydrological processes are very complex, and it is possible that they will never be comprehended in its totality. However, a simplified approach is to assess it as a system. Systems are defined as a set of connected parts that form a whole. Hydrological systems are the consist on the representation of such complex dynamics through connected equations. Each equation models single or multiple mechanisms pertaining a single cycle. Each cycle takes place at a different compartment and is propagated to other compartments, with the idea of forming also, an entire system(Chow et al.,1988). Figure 3 shows a system scheme for the rainfall-runoff process.

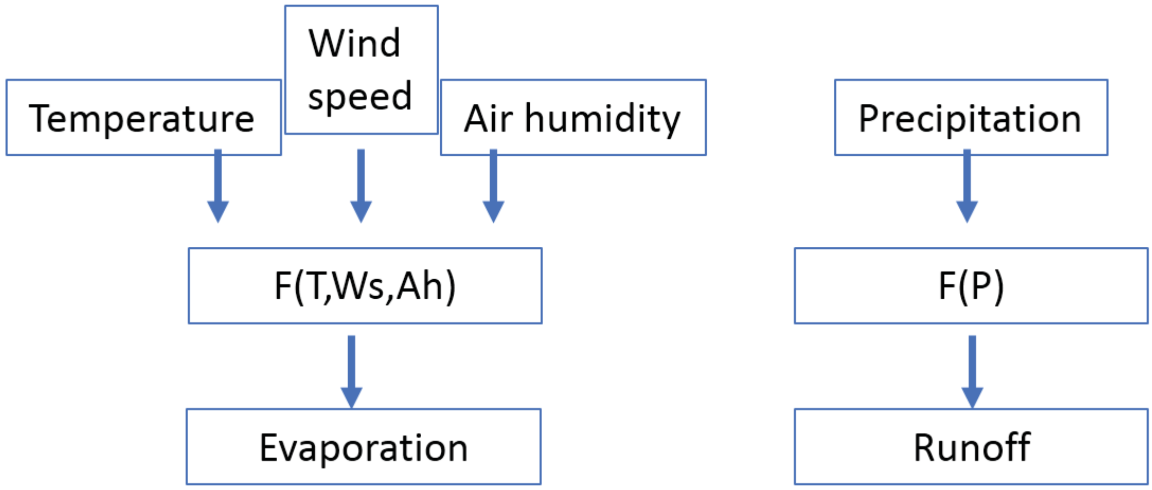


Figure 3 – Representation of two systems of the hydrological cycle

Source: CHOW et al. (1988) – Adapted

Diffuse pollution is a major cause for water quality loss. It originates from various sources, as different types of activities and land covers are performed, either in rural or urban environments. Agricultural activities are known sources of fertilizers and pesticides. Farming practices frequently employ fertilizers, which are sources of macro nutrients (nitrogen and phosphorus). Such substances can cause eutrophication and negatively affect aquatic biota. Nutrient paths on the environment are: (i) transport by runoff wash off, (ii) evaporation followed by deposition and (iii) soil and groundwater infiltration. Fertilizers are relevant sources to aquatic bodies, if excessively applied and causing nutrient surpluses in the soil. Pesticides are subject to transport by (i) wind, (ii) surface runoff, (iii) groundwater. Different agricultural operations, such as irrigation, row cropping and monoculture also influence soil loss and nutrient export, as uncovered land is more susceptible to runoff and wind transport (NOVOTNY ,1999). Cattle farming is another important source of diffuse pollution on rural environments. As manure production increases with the livestock density, areas required for manure application, relative to Confined Animal Feeding Operation (CAFO) area also are significantly larger (CARPENTER et al., 2008).

In urban environments, a more predominantly heterogeneous set of sources inputs different substances to aquatic bodies either through deposition, runoff scouring along natural pathways or conveyance channels. Tire and brake dust, oil and fuel spills on the asphalt contribute respectively to increase of Total Suspended Solids(TSS) and Heavy metals (Pd, Cd, Cu) (FUKUZAKI et al., 1986). The USEPA(USEPA., 1999) and Barrett et al. (1998) report considerable load values of TSS, Pb, and Cu in highway and freeways, comparable to industrial and commercial land uses. Another important source in urban areas is roof runoff. Its stormwater pollutants are originated from wet and dry deposition of atmospheric particulate matter. It is influenced by meteorological parameters (rainfall intensity, wind speed and direction, antecedent dry days), roofing material (type, age and slope) (ZOBRIST et al., 2000; CHANG et al., 2004). Tobiszewski et al. (2010) analyzed stormwater sampled in different roof structures and found TSS, Nitrites, nitrates, Cadmium, Zinc, Cooper, pesticides and other organic compounds. Table 1 shows average concentration of 8 parameters for urban stormwater and domestic raw sewage. It was obtained from the Nationwide Urban Runoff Program preliminary results report, conducted by the USEPA.

Table 1 - Average concentrations for Raw Sewage and Urban runoff

Constituent	Urban runoff		Domestic Wastewater		
	Separate sewers		Before treatment		After secondary treatment
	Range	Typical	Range	Typical	Typical
COD (mg/L)	200-275	75	250-1,000	500	80
TSS (mg/L)	20-2,890	150	100-350	200	20
TOTAL P(mg/L)	0.02-4.30	0.36	4-15	8	2
TOTAL N (mg/L)	0.4-20.0	2	20-85	40	30
Lead (Pb)(mg/L)	0.01-1.20	0.18	0.02-0.94	0.1	0.05
Copper (Cu)(mg/L)	0.01-0.40	0.05	0.03-1.19	0.22	0.03
Zinc (Zn)(mg/L)	0.1-2.90	0.2	0.02-7.68	0.28	0.08
Fecal Coliform per 100 mL	400-50,000		10^6-10^8		200

As depicted in the above, runoff concentrations for some pollutants, such as solids and metals can reach concentrations similar to domestic sewage. Hence, mitigating the impacts of diffuse pressure on surface waterbodies is important for aquatic environment conservation and restoration efforts. The monitoring and assessment on nonpoint pollution effects in watersheds is still a recent field in many countries, including Brazil (YANHUA et al., 2012). Diffuse loads are sources source of waterbody impairment in many locations throughout the world(DE OLIVEIRA ET AL., 2017). Management strategies are efforts that include but are not limited to integrated legal framework, modelling and structural and nonstructural management practices to reestablish aquatic life-favorable mass balances.

3.2 Hydrological quantity and quality control measures – Structural and nonstructural

Land use changes in Brazilian Urban spaces over the last decades has caused increase on runoff quantities and flow, due mainly to land use changes and impermeabilization in various urban centers. As a direct consequence, flooding issues arisen. The adopted solution then was building and expanding the existing conveyance systems, besides dredging, replacing natural with constructed courses with and rectifying natural channels. However efficient these solutions remain and their well-known benefits, it does change aspects of the hydrological cycle which are not beneficial for the users, such as affecting infiltration and causing lower aquifer baseflows and consequently lower channel flow during dry periods (CANHOLI, 2015).

However direct recharge is reduced by the impermeabilization of surfaces, research suggests that underground system leakages lead to larger recharge rates than when compared to natural surfaces(LERNER, 2002). The modelling of infiltration and groundwater process is extremely complex and frequently difficult to gage due to costs associated with data availability.

The policies adopted in Brazil concerning the adoption of flood control also does not addresses nonpoint pollution, which tends to send water with worse quality issues to the lower watershed regions. One of the instruments currently employed in Brazil to control river pollution is the classification (*enquadramento*), a federal regulation that categorizes rivers and sets maximum pollutant thresholds for several constituents (BRASIL, 2005). State agencies are required to conduct framing of in-state waters and enforce correspondent standards. However, no diffuse pollution control measures prior to waterbody discharge are required. Also

Despite the success and benefits of this constructed solutions approach, alternative solutions on urban drainage have been developed to control stormwater flow and pollution and diminish watershed impact. Despite similar terminologies and goals, they differ in their approach and history. These are namely: (i) the Low impact development(LID) approach, which comprehends practices mainly implemented on site to control stormwater at source and restore natural hydrologic processes and pre-development conditions, (ii)Green Infrastructures(GIs), which is very similar to LID but includes green space networks spaces and corridors with objectives beyond stormwater management(e.g.: maximizing ecosystem services, watershed restoration, and biodiversity)(ZHANG; CHUI ,2018) (iii) Best Management Practices (BMP) approach, developed in the United States. The latter, according to the U.S. code of law(UNITED STATES FEDERAL LAW, 1962), it aims to improve surface waters, through the Total Maximum Daily Load(TMDL) permitting program. BMPs include but are not limited to structural or nonstructural measures. (iv) Sustainable Drainage Systems (SuDS), approach created in the United Kingdom, comprehends a philosophy that employs structural solutions to slow down and reduce the quantity of surface water runoff from a developed area, and reducing risks of runoff pollution (KELLAGHER et al., 2015;WOODS-BALLARD et al., 2007). Both approaches utilize structural solutions to achieve volume, flow and quantity improvement.

These terminologies will be used interchangeably in this work, always referring as BMP as constructed devices to control flow and water quality.

Many of these devices may be used for recreational purposes (e.g. playgrounds), or aesthetical amenities for surrounding communities (i.e.: lakes, ponds and wetlands). Figure 4 shows a constructed wetland in the United States. The wetland serves as a park and community space for recreation.

Figure 4 – Tanner springs Park in Portland, OR

Source:https://www.revelstokemountaineer.com/a-new-kind-of-park-for-revelstoke/tanner-springs-park-oregon/

In any approach, both structural and nonstructural solutions are employed as a part of stormwater management plans, which aim to preserve or restore water quality and quantity conditions. Structural BMPs may be used isolated or as parts of a treatment train, having underground discharge as main outlet path (PORTLAND ,2016). BMPs are capable to interfere in different phases of stormwater wash off: source-control, in-situ controls and regional controls

(WOODS-BALLARD et al., 2007). Figure 5 depicts a few examples of urban BMPs and a summary on mechanisms and known effects.

Many more BMPs devices are available, with different goals, drainage areas and flow and water and quality performances and mechanisms. Wetlands (Shallow wetland, pond/wetland), infiltration devices (infiltration trenches and soakways), filtration devices (surface and subsurface sand filters, bioretention/filter strips), detention basins and open channels (wet and dry swales) are examples of stormwater management devices well known and documented for stormwater control. The applicability of a certain structural solution is dependent in many factors, that include but are not limited to: (i) site-specific conditions, (groundwater levels, land availability, soil types, etc.), (ii) receiving stormwater current quality, (iii) desired treatment removal, (iv) desired flow and volume abatement and (v) implementation and operation and management costs. Devices may be employed at various effect scales, namely source controls, in situ-treatment and regional treatment, either isolate or as part of a treatment train (KELLAGHER et al., 2015).

Best Management Practices are mainly sized according to its main goal. It may be whether flow control or quality control. Within flow control, different criteria may be adopted to design devices. For example. BMPs may be sized to accommodate a certain design volume during a certain detention or retention times. Various manuals and compilations are available from several Federal (USEPA) and State Agencies, and provide guidelines for stormwater design, usually preceded by the law framework and the flow and water quality requirements for various permit processes.

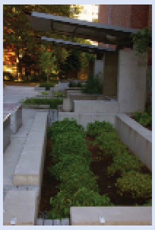
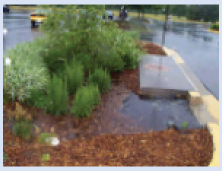
	Description	Flow reduction	Pollutant removal and mechanisms	Uses
 Infiltration basins	Detention basin with succulent vegetation	Infiltration Retention	TSS, N,P metals • Sorption • Interception	Street/building runoff
 Biorretention	Engineered garden with succulent vegetation	Infiltration Retention Detention	TSS, N,P metals • Sorption • Interception	Block scale runoff treatment
 Swales	Vegetated shallow v-shaped conveyance channel	Infiltration Detention	TSS • Sorption • Interception • Plant uptake	Street/road runoff
 Vegetated roofs	Multi-layer porous media bed vegetated with succulent vegetation	Detention	-	Rooftops

Figure 5 - Examples of BMPs, main applications and functional mechanisms

Source: Kellagher et al. (2015), adapted

Different responses from the land phase of the hydrological cycle are observed in urban and rural land uses. In non-urban environments, infiltration occurs more homogeneously, in addition to higher runoff/precipitation rates. This behavior associated with larger concentration times grants a “rural watershed behavior”, where in-event peaks are attenuated, baseflow recession is less steep and the dry-period flow is higher. In contrast, in urban areas, flow tends to be larger due to larger runoff quantities as impervious area decreases. Also, the flow is accelerated by the usage of conveyance systems, which drastically reduce the concentration times, causing higher peak flows and also faster recession periods. This behavior is classified as a “urban watershed behavior”. Figure 6 shows illustrates both concepts using event hydrographs.

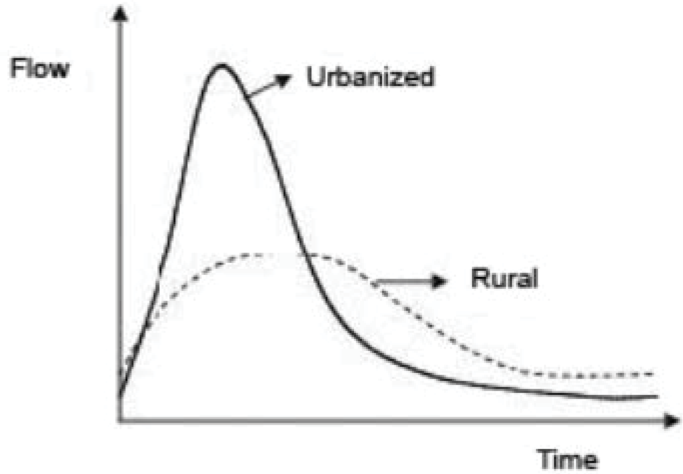


Figure 6 - Rural and Urban watersheds

Source: TUCCI (2010)

As direct effects of the changes in land use in rural environments, top-soil layer changes, decreased evapotranspiration and increased erosion are some of the effects of natural land conversion. As for urban environments, decreased evapotranspiration, decrease of solids interception and soil compaction are some of the observed effects (MAIDMENT,1993)

In terms of water quality, in rural environments, the capture of substances on the environment depends on management practices that control the export of substances along different stages of crop growth, such as manure application and crop rotation. Manure application (e.g.: barnyard runoff control, optimization fertilizer application), agricultural composting and chemical fertilizer BMPs are examples of processes that can reduce sediment transport and chemical scouring. Diversification of crops (crop rotation) utilizes residual nitrogen are also effective measures. As for agrochemical application, timing of application, proper quantity application and proper maintenance on application equipment are examples of BMPs (MERRILL et al., 2012). In urban environments, land use and land cover (LULC) planning, preservation of buffer areas for sediment entrapment, street sweeping, household hazardous chemicals proper disposal, impact, lawn mowing, and fertilizer application are examples of non-structural BMPs. They also comprehend public education outreach programs targeted for: (i)the general population (i.e.: the effects of littering) and (ii)Business (i.e.: Carwashes, Industries)(USEPA, 2017)).

3.3 Quantity and quality Watershed modelling: Introduction and modelling techniques

According to Mignot et al. (2019), hydrological models involving urban hydrological analysis depends severely in the quantity of data available. The modelling of urban surfaces, pathways and features are very complex and diverse. In terms of availability of models, its complexity may be 1D, 2D and 3D approaches. Also, the choice of modelling either steady-state of transient-state flow besides adding complexity also induces uncertainty on modelling. Also, as flow leaves river stream, routing processes become harder to model as velocity fields and spatial resolution becomes more complex.

Watershed nonpoint pollution on superficial waters is the result of a series of processes that occur inside watershed limits. It depends on climate conditions and on watersheds response to precipitation events. These processes are heterogeneous in time and space and establishing relationships between physically measurable parameters and a realistic representation is challenging. Quantification of the involved mechanisms depends on climate, discharge, and water quality monitoring. The latter describes watersheds environmental processes lumping them in the form of water quality parameters measured from samples. Each sample describes a system in one single location, at a single time (i.e.: sample collection and water quality parameter quantification).

Such parameters are used to assist watersheds mass processes and when coupled to algorithms and models, simulate different scenarios and assist decision-making process. Models are mathematical descriptions of a real system, created to account for the effects measured on real conditions. They are either based on (i)physical relationships (i.e.: water and constituent mass balances) or (ii)conceptual relationships (i.e.: manning equation, Penman Method, and partitioning coefficients), or a combination of both. Models can be also account (distributed models) or not (concentrated models) for space variability. Both hydrological and water quality models are subject to various levels of uncertainty (i.e.: input parameter values, accurate process representation).

Another important classification about water quality models concerns its reference system, as they can be divided in: (i) receiving water models (model the behavior of conveyance systems: rivers, estuaries, etc.) and (ii)watershed models (emphasize in describing hydrology, pollutant transport, erosion and other processes). It is true that not all surface water quality problems are caused by diffuse sources. Furthermore, not all the diffuse pollution cases require watershed simulation for water quality assessment.

Many models and techniques for evaluating diffuse pollution are available. One way to divide efforts concerning the spatial identification of diffuse pollution, and the one used to guide this review is: (i) Diffuse pollution ranking techniques (ii) Spatial location, sizing and benefit accounting, (iii) Decision support systems for BMP implementation/sizing/benefit accounting and (iv) Watershed Modelling and physical/conceptual approaches.

This classification is merely for the purposes of organizing a review. None of the efforts is categorically implemented alone as many studies seek to join various cost-benefit, environmental impacts, social costs and other important characteristics to assist decision making. Another model distinction between models is proposed by Lane et al. (2006), distinguishes models according to computation principles as: (i) Inference functions: correlation functions between and observed data relationships are sought (ii) Mass transfer functions: Functions that employ mass export rates and mass export models. (iii) Land-Soil modelling: Physical and conceptual laws are applied to model pollutant transport (iv) Soil transfer modelling: combination of (3) with dynamic algorithms to account for non-constant mass inputs (i.e.: tillage practices, pesticide application). Models also may or may not contain physical, conceptual and in a smaller group, stochastic basis, assisting specifically the decision-making process.

A very popular approach to the diffuse pollution source area ranking is export coefficient methods. These methods have been employed successfully to quantify land use impact on surface waters (OMERNIK ,1976 ; BORAH; BERA, 2004). One example is the Pollutant Loading Estimator (PLOAD) model, developed by the USEPA, estimates diffuse pollution loads based on export coefficients and has been successfully employed in studies (SHI et al., 2012;GURUNG et al., 2012; SHEN et al., 2011; LIN et al., 2016). Another approach is composing indexes for diffuse pollution ranking. The Potential Nonpoint Pollution Index(PNPI)(MUNAFÒ et al. (2005) is a GIS-based critical source area ranking method that uses three indexes: (i)Land Use index, (ii) Runoff Index and (iii) Distance from stream index. Within that same approach, another well utilized algorithm is the Sensitive Catchment Integrated Modelling Analysis Platform (SCIMAP), developed in the United Kingdom, evaluates diffuse pollution risks throughout an hydraulically connected network within a GIS environment, with its indexes selected by experts(MILLEDGE et al., 2012). The Agricultural Potential Nonpoint Pollution index(APPI) (PETERSEN et al., 1991) ranks areas employing four indexes: runoff, sediment production, chemical use and livestock population index. Other ranking methods such as the one proposed by Koo; O'Connell (2006) also consider social and economic aspects.

Several algorithms for ranking Critical Source Areas (CSAs) or Hydrologically sensitive areas (HSAs) have been developed by coupling pollutant fate and transport models to evolutionary optimization algorithms and will be discussed separately in section 3.4 of this document, as they relate to simulation models.

Another category are watershed simulation models. These models employ rainfall-runoff relationships associated to water quality models. Examples are the Soil and Water Analysis tool (SWAT), Hydrological Simulation Program-Fortran (HSP-F), Mike-SHE and the Stormwater Management Model (SWMM)(LI et al., 2014). These models compute various environmental processes over time, under various data quantity and resolution requirements. They must be able to represent site hydrological and water quality conditions. Site-specific data is necessary to assess pollutant export and its environmental effects. Models can quantify water quantity and quality enhancements of structural and non-structural watershed management measures, such as BMP, LID and GI infrastructure. However, not always is possible to establish hydrological and water quality data networks enough to satisfy simulation requirements. Frequently data imported from other flow and climate stations must be imported for hydrological assessment, which also consists in a source of uncertainty (WANG; KALIN, 2011). Monitoring programs are frequently costly and labor intense(FARZIN; KAPLAN, ,2004). This sampling approach has the downside of condensing many environmental processes into single snapshots in space and time, transforming them in measurable parameters (WARDROPPER et al., 2017). Thus, required sampling spatial and temporal resolutions become a key issue on establishing sufficient and representative data to effectively achieve monitoring program goals.

Many studies have been conducted on employing nonpoint pollution models to unobserved or scarcely observed watersheds (ÖZCAN; BAŞKAN; et al. ,2017; ÖZCAN et al., 2017 ; PANAGOPOULOS et al., 2011, ROMAGNOLI et al., 2017 ; BRANNAN et al., 2014). These studies are based in literature values and subsidized most times by local databases, such as DEMs, and indirect measurement techniques such as hyperspectral image processing (LIN et al., 2015) or LiDar imagery (WILLIS et al., 2017). Even though adequate monitoring programs provide a better watershed assessment, research has shown good results on identifying runoff flow and mass transport processes. Harmel et al. (2014) divides the uses of diffuse pollution modelling results in: (1) Exploratory, (2) Planning and (3) Regulatory/legal. Each use requires acceptable data inputs to ensure a correct representation of a real scenario.

Simulation models became increasingly popular as computation capability allowed and GIS collaboration systems were developed to assist either the data entry processes or as model structure itself, in the case of models based on analyzing on geo-objects (LI et al., 2014) (i.e.: map algebra on models such as the PNPI). Most of the simulations models are legacy code, and hence the focus of model development has been on integrating GIS systems to enhance data input, (i.e.: watershed delineation, parameter data extraction and sub basin unit division) for model inputs (BORAH; BERA, 2004). Consequently, uncertainty issues concerning resolution and quality of georeferenced data input may arise. A correct representation of a watersheds geographical and morphological structures is a function of topographical and land use dynamics complexity and required precision for model input (ERTURK et al., 2007). Land use, soil maps, and Digital Elevation models are common inputs in diffuse pollution models. Shen, et al. (2013) compiled several results for Digital Elevation models (DEM) employing the Soil and Water Analysis tool (SWAT) and concluded that there is a threshold to which DEM resolution increase affect hydrological cycle and water quality representation. Xu et al. (2016) reports significant differences on data inputs for critical diffuse pollution area tagging using SWAT. Chen et al. (2016) and Geza; McCray (2008) report no substantial changes with increasing soil data detail. In conclusion, the definition of which model or how to employ a model for diffuse pollution assessment is an exercise of balancing various data sources and expectations for various interested parties.

Pollutant fate and transport models represent more or less thoroughly constituent sourcing and hydrological processes, usually performing better a certain LULCs, soil, climate conditions and simulation timesteps. Currently operational models (i.e.: models featured with detailed user manuals, online support communities and currently under development) are to an extent rather suitable for agricultural or urban conditions.

In rural watersheds, infiltration, water withdrawal, and agricultural practices are important to water balances. Manure, pesticide and soil management practices (e.g.: tillage, cover crops), are more likely to contribute to chemical and sediment transport processes, exerting heavier influence on the system.

In urbanized watersheds, flow is transferred between pervious and impervious surfaces, soils are significantly compacted, and channelized flow gains significance on the mass transfers to the main streams. Groundwater recharge becomes less important feature to be represented, as soil compaction impedes lower infiltration. In a Storm Separate System (SSS), the case of the Barigui Basin, stormwater conveyance channels become more relevant sources to flow (since concentration times are reduced), accentuated by channel infiltration due to cracks and manholes. Channel straightening, channelization and flood containment measures such as ponds, or retention tanks become of interest to represent on the water balances. In terms of water chemical speciation, impervious surfaces are main pollutant sources, settled through wet and dry deposition or deposited by human action. In terms of hydrograph response, the smaller response times are required smaller, decreased interval computation to adequately simulate its effects(DAI et al., 2017).

In certain cases, water balances can be of hard representation. Semi-arid and arid climates, as well as specific groundwater arrangements such as fractured Karst (AMIN et al. ,2017 ; MALAGÒ et al. ,2016 ; ÖZCAN, KENTEL, et al., 2017) are hard to represent in terms of water yield and release. Hence, proper accounting for these processes is not available in all models and may be application limited.

Adequate timestep for system description, availability of data, implementation time, computational availability and required processes to be reproduced are some of the main factors to be taken into consideration when choosing a diffuse pollution model. As data acquisitions cost rise, and a balance of interested parties must be met. A study plan must be established to determine an acceptable modelling strategy. An overview of three widely used hydrological/pollutant and fate transport is presented next:

The Agricultural Nonpoint Source Pollution (AGNPS) is an event-based model designed to estimate loads from agricultural management schemes. It simulates surface runoff, nutrients, sediments, chemical oxygen demand (COD) and pesticides from point and nonpoint sources. The program employs a grid-based watershed representation, and it simulates rural BMPS. The Hydrology module is based on the SCS Curve Number Method(NRCS ,1986) and the sediment loss is computed using the Universal Soil Loss Equation(USLE). It is a model recommended for use in watersheds with areas superior to 80 mi^2 (207.20 km^2). Among its limitations, it assumes channels have triangular shape and its empirical hydrological process representation. The AnnAGNPS is a continuous version of AGNPS. It computes homogeneous drainage areas (HDUs). 1D continuous channel flow is considered and a weather generator is available.

The Hydrological Simulation Program – Fortran (HSP-F) is a model for watersheds based hydrologically on the Stanford Hydrological method and in the ARM (Agricultural Runoff Method). It simulates HRU based grids in hourly up to daily timesteps. The model assumes a HRUs (Hydrological Response units), lumping water balance at the sub watershed in each timestep. The HSPF is a part of the Better Assessment Science Integrating point and Nonpoint Sources (BASINS), a platform that provide a single GIS integrated input/output and for various models, including food chain, surface water quality and hydrology.

The Stormwater Management Model (SWMM) is a model applied primarily to urban areas for single-event or continuous simulation. SWMM is capable of simulating full dynamic wave routing, in which it can simulate nonlinear reservoirs, pressure flow and backwater scenarios. Infiltration is calculated using the Horton or Green-Ampt methods. A modified version of Manning equation is used to compute flow rate from sub catchment areas as nonlinear reservoirs. Groundwater and soil storage are represented by a lumped storage scheme. SWMM is capable of computing sediment using the USLE or other urban built-up and wash off models. SWMM has limited capability to simulate soil and groundwater interactions as well as non-urban environments.

The Soil and Water analysis Tool (SWAT) is a hydrological and water quality model BMP simulation has been studied for years. Sedimentation-filtration basins, ponds, wetlands, and agricultural practices are available (Neitsch et al., 2011). Sub-daily and sub-hourly scaling is still an object of development at a number of institutions including Texas A&M university itself (ARNOLD; FOHRER, 2005 ; WAIDLER et al., 2011).

Ferreira et al. (2018) compiled diffuse pollution models and analyzed for applicability in watersheds in Brazil, according to degree of user support, time of implementation and published literature. LID representation features are completed by the revision from Bosley et al. (2008) and depicted in Table 2.

Table 2- Diffuse pollution model compilation

			SWMM	SWAT	HSPF	AGNPS	AnnAGNPS	CE-QUAL-W2	MIKE SHE	HEC - HMS	STORM
BASIC FEATURES	MODEL NAME / ACRONYM		SWMM	SWAT	HSPF	AGNPS	AnnAGNPS	CE-QUAL-W2	MIKE SHE	HEC - HMS	STORM
	DEVELOPER/DISTRIBUTER		USEPA	Texas A&M University	USEPA	USDA-ARS	USDA-ARS / USDA-NRCS	USACE	DHI	USACE	USACE
	Level of Complexity	Export Coefficients	-	-	-	-	-	-	-	-	✓
		Loading Functions	-	-	-	-	-	-	-	-	-
		Phisically-Based	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Land Cover simulation capability	Urban	✓✓✓	✓✓	✓✓	-	-	-	✓✓✓	✓✓✓	✓✓✓
		Rural	✓✓	✓✓✓	✓✓✓	✓✓✓	✓✓✓	-	✓✓✓	✓✓✓	-
		Agriculture	✓	✓✓✓	✓✓✓	✓✓✓	✓✓✓	-	✓✓✓	✓✓✓	-
	✓ Simplified/ limited	Forest	✓	✓✓✓	✓✓✓	-	-	-	✓✓✓	✓✓✓	-
	✓✓ Medium	River	✓	✓	✓✓✓	-	-	✓✓✓	✓✓✓	✓✓✓	-
	✓✓✓ High	Lake	✓	✓	✓✓	-	-	✓✓✓	-	-	-
		Reservioir/Impoudment	✓✓	✓	✓✓	-	-	✓✓✓	✓✓✓	✓✓	-
	Orientation	Event-oriented	✓	-	-	✓	-	-	✓	✓	✓
		Continuous	✓	✓	✓	-	✓	-	✓	✓	✓
	Data quantities	Low	-	-	-	-	-	-	-	-	-
		Medium	-	✓	-	✓	✓	-	-	-	✓
		High	✓	-	✓	-	-	✓	✓	✓	-
	Operational	Yes	✓	✓	✓	✓	✓	✓	✓	✓	-
		No	-	-	-	-	-	-	-	-	✓
	Timestep	Sub-Daily	✓	✓	✓	✓	-	✓	✓	✓	✓
		Daily	-	✓	-	-	✓	✓	✓	-	-
		Monthly	-	-	-	-	-	-	-	-	-
	Required modelling experience Required (1 to 4)		✓✓✓✓	✓✓✓	✓✓✓✓	✓✓	✓✓	✓✓✓✓	✓✓✓✓	✓✓	✓✓✓
	GIS Compatibility		✓	✓	✓	✓	✓	-			
	Time Needed for Application		< 3 Months	> 1 Month	> 6 Months	> 1 Month	> 1 Month	> 6 Months	> 6 Months	> 1 Month	< 1 Month
	Cost		Public Domain	Public Domain	Public Domain	Public Domain	Public Domain	Public Domain	Significant cost	Public Domain	Public Domain
	Programming language		C	Fortran	Fortran	Fortran	Fortran	Fortran	*	Java/Fortran	*
	References		USEPA, 2015	S.L. Neitsch et al., 2009	USEPA, 1993	*	Binger et al., 2015	Cole, T.M., and S. A. Wells. 2003.	Abbott, M.B et al(1986)	USACE,2003	USCE, 1977
COMPUTATIONAL FEATURES	Point sources		✓	✓	✓	✓	✓	✓	-	-	-
	Groundwater transport		✓	✓	✓	✓	✓	-	✓✓	-	-
	Reservoirs/impoudments(1 to 3)		✓✓	✓	✓✓	-	-	✓	✓✓✓	✓✓	-
	Sediment transport		✓✓	✓✓	✓✓	✓✓	✓✓	-	✓✓	-	-
	Field Practices		-	✓	✓	✓	✓	-	✓	-	-
	BMP Computation		✓	✓	✓	✓	✓	-	✓	-	✓
	BMP representation	Generalized	-	-	✓	✓	✓	-	-	-	✓
		Detailed	✓	✓	-	-	-	-	✓	-	-
	Types of BMPs	Detention Basins	✓	✓	✓	✓	✓	-	*	-	-
		Urban Infiltration Practices	✓	✓	✓	✓	✓	-	*	-	-
		Urban Vegetative Practices	✓	✓	✓	✓	✓	-	*	-	-
		Wetlands	✓	-	-	-	-	-	*	-	-
		Others	✓	✓	-	-	-	-	*	-	✓

	NAME		SWMM	SWAT	HSPF	AGNPS	AnnAGNPS	CE-QUAL-W2	MIKE SHE	HEC - HMS	STORM
MATHEMATICAL BASIS	Watershed representation	Homogeneous pervious/imp. areas	✓	-	-	-	-	-		-	*
		Square grids with uniform hydrologic characteristics	-	-	-	✓	✓	-	*	✓	*
		Pervious and impervious land areas, streams, channels and reservoirs	-	-	✓	-	-	-	*	-	*
		Square overland grids, 1-D channels, 3-D saturared-flow layers	-	-	-	-	-	-	*	-	*
		Hydrological Uniform Units(HRU)	-	✓	-	-	-	-	*	-	*
	Runoff Model		Philips equation	CN method/Green-Ampt Mein Larson	Maximum Surface Retention	CN method	CN method	-	Richards equation	Unit Hydrograph/SCS CN/Green-Ampt	Coefficient method/SCS CN
	EVT Model		Hargreaves /time series	Hargreaves/ Penmann	Empirical Relationship	Penmman Equation	Penmman Equation	-	Penman-Montieth/Kristensen-jensen	User Supplied	Empirical relationship
	Channel Routing		Dynamic Wave/Kinematic Wave/Continuous flow - predefined shapes or user-defined	Manning/ Trapezoidal	Manning/ Chezy	Manning	Manning	Full Saint-Venant equations	Muskingum /diffuse wave	Muskingum, Muskingum-Cunge, Puls mod, Kinematic wave	SCS triangular UH
	Sediment Computation	MUSLE		✓		-	-	-	*	✓	-
		USLE		✓		-	-	-	*	-	✓
		Toffaleti/Colby Equation			✓	-	-	-	*	-	-
		RUSLE		✓		✓	✓	-	*	-	-
OTHERS	Simulation of land use/land cover scenarios		✓	✓	✓	✓	✓	-	✓	✓	✓
	GIS Compatibility		✓	✓	✓	✓	✓	-	✓	✓	✓
WATER QUALITY CONSTITUENTS	User-Defined Constituents		✓	-	✓	-	-	*	*	-	-
	Nitrogen	Ammonia (NH3) Concentration	✓	✓	✓	-	-	✓	-	-	-
		Nitrate (NO3) Concentration	✓	✓	✓	-	-	✓	-	-	-
		Total nitrogen Concentration	✓	✓	✓	✓	✓	✓	-	-	✓
		Total nitrogen Load	✓	✓	✓	✓	✓	✓	-	-	✓
	Phosphorous outputs	Total phosphorous Load	✓	✓	✓	✓	✓	✓	-	-	-
		Total Phosphorus Concentrations	✓	✓	✓	✓	✓	✓	-	-	-
	Sediment outputs	Sediment Load	✓	✓	✓	✓	✓	-	-	-	✓
		Sediment Concentration	✓	✓	✓	✓	✓	✓	-	-	✓
	TSS outputs	TSS load	✓	✓	✓	-	-	✓	-	-	-
		TSS Concentrations	✓	✓	✓	✓	✓	✓	-	-	✓
	Metals outputs	Water	✓	✓	✓	-	-	*	-	-	-
		Sediment	-	-	✓	-	-	-	-	-	-
	Pesticides Processes	Land Phase	-	✓	✓	✓	✓	✓	-	-	-
		Conservative Constituents	-	✓	✓	✓	✓	✓	-	-	-
		Partitioning reactions	-	✓	✓	✓	✓	✓	-	-	-
	Oxygen Balance	Dissolved Oxygen	-	✓	✓	✓	✓	✓	-	-	-

3.4 Quantity and quality pollution modelling – SAOT's and watershed modelling

BMP implementation at field-scale is subject to physical and climate site conditions, stakeholder interests, and watershed benefits. To tackle this issue, spatial allocation optimization tools (SAOTs) have been proposed as solutions in the literature. These tools use indicators and optimization tools to optimize LID location and performance, coupling diffuse pollution source tagging (whether through physical models, or watershed models), and multi-criteria and or evolutionary algorithms. The problem formulation starts with optimizing one BMP or BMP treatment trains, changing spatial optimization based on real-site constraints and BMP sizing. Objective functions may include other BMP life cycle periods such as cost, social benefits or account for flood reduction risks. SAOTs are mainly constituted by a (i) Parameter generator (ii)calculation engine (iii)Decision-making tool that evaluates results. The latter may generate future estimates for future iterations in evolutionary models. Normally the decision is optimized based on hydro-environmental performance (peak reduction, costs)(ZHANG; CHUI, 2018). Many SAOTS are available to this day, with varying methods to account for BMPs in water balances and optimize BMP allocation and sizing.

Using SWAT as a computation engine, Bekele; Nicklow (2007) applied a Non-dominated Sorting Genetic Algorithm II (NSGA-II) to calibrate flow for a 133 Km^2 watershed using the Root Mean Square Error (RMSE) as an efficiency indicator. The first application of the NSGA algorithm for spatial allocation was done by Kaini et al. (2007), using a GA algorithm to minimize stream sediment delivery and minimizing BMP construction cost in a 1,189 Km^2 watershed. The study applied detention ponds, filter strips and grade stabilization, re- using the pre-built SWAT routine for and filter strips and re-parameterizing HRUs for the others. Maringanti et al. (2009), optimized farms modelled as individual Subbasins for pollutant BMP removal and costs reduction associated with sustainable practices. Shen, Z. et al. (2013) developed the topography analysis incorporated optimization method (TAIOM), which uses a GA coupled to a multi-objective function, computing pollutant loads, a surface indicator and, slope indicators and construction costs for a 47.24 Km^2 for a watershed in China testing for four BMPs(wetlands, detention ponds, vegetative filters and grassed waterways More recently Dai et al. (2017) optimized a Fuzzy credibility chance constraint for a 1029 Km^2 watershed in China for parallel terracing, grade stabilization, filters strips and detention wetlands. Other studies are available concerning the application of evolutionary algorithms.

SWMM is the most utilized model as calculation tool for SAOTs. Currently Genetic algorithms (GIACOMONI, 2017; SEBTI et al., 2016), Particle Swarm optimization (PSO) (DUAN et al., 2016) and TOPSIS, a multi-criteria non-evolutionary BMP allocation tool(SONG; CHUNG, 2017). Many more evolutionary algorithms are available using various computational engines, such as the AnnAGNPS(SRIVASTAVA, 2003) and SUSTAIN(MAO et al., 2017). This work does not focus in optimizing BMP sizing, allocation and distribution, and this is intentionally a brief review, conducted through researching the most relevant articles related to the subject in the ScienceDirect database and correlated studies.

3.5 The Soil and Water Analysis Tool: introduction & equations

The Soil and Water Analysis tool is a model is developed by the Texas A&M university. It is an operational, physically based model, that computes land and channel phases in daily and sub daily timesteps. The inputs for SWAT are precipitation, maximum and minimum air temperature, solar radiation, wind speed and relative humidity (NEITSCH et al., 2011). Channel and subbasin parameters are also inputs and can be obtained from GIS data. In the model, a watershed is divided in Subbasins. Each subbasin has Hydrological Response Units (HRUs): discrete units with similar hydrological responses, obtained by combining (1) Land use and land cover, (2) Soil type and (3) Slope categories. HRU definition is based on: (i)minimum area to constitute an HR. Case contrary is lumped to the closest dominant HRU combination, (ii)maximum number of HRUs per subbasin or (iii)formation of one HRU based on the dominant combination of LULC, soil and slope in the subbasin.

The ArcSWAT Extension is a plugin for the ArcGIS platform. It assists data extraction from georeferenced databases, pre-processing the input files (WINCHELL et al., 2013). It also aids climatic data input, HRU, subbasin and watershed delimitation, point source input creation parameter estimation, among other functionalities. ArcSWAT compiles information using .mdb tables using in a SQLite ODBC engine. Subsequently, it writes input text files for model run.

SWAT simulates sub daily, daily, monthly and annual timesteps. Equation 1 show the daily (monthly and annual share the same equations) water balance equations. Equations for sub-daily timesteps are modified for enhanced representation of more time-sensitive processes.

$$SW_t = SW_0 + \sum_{i=1}^{t}\left(R_{day} - Q_{surf} - E_a - w_{seep} - Q_{gw}\right) \qquad \text{Equation 1}$$

Where $\boldsymbol{SW_t}$ is the final soil water content(mm), $\boldsymbol{SW_0}$ is the initial soil water content on day i(mm), t is the time(days), $\boldsymbol{R_{day}}$ is the amount of precipitation on day i(mm), $\boldsymbol{Q_{surf}}$ is the amount of surface runoff on day i, $\boldsymbol{w_{seep}}$ is the amount of water entering the vadose zone from the soil profile on day i (mm) and $\boldsymbol{Q_{gw}}$ is the amount of return flow in day i. SWAT uses for runoff separation the SCS curve number method NRCS (1986). The equation for $\boldsymbol{Q_{surf}}$ is below in Equation 2:

$$Q_{surf} = (R_{day} - 0.2S)^2 / (R_{day} + 0.8S) \qquad \text{Equation 2}$$

Where $\boldsymbol{Q_{surf}}$ is the accumulated runoff excess in the time t (mm), $\boldsymbol{R_{day}}$ is the accumulated daily rainfall (mm), and $\boldsymbol{S}$ **i**s the retention parameter (mm).

The CN value is calculated according to Equation 3:

$$S = 25.4\left(\frac{1000}{CN} - 10\right) \qquad \text{Equation 3}$$

Where **CN** is the Curve Number for a given day. CN values under dry conditions are estimated based on the CN2 value, a simulation initial value, and is calculated as shown in Equation 4:

$$CN1 = CN2 - \frac{20*(100-CN2)}{100-CN2+\exp[2.533-0.0636(100-CN2)]} \qquad \text{Equation 4}$$

CN3 values, which correspond to the CN under wet conditions are estimated based on the CN2 value, and is calculated as described in Equation 5:

$$CN3 = CN2 * \exp[0.0673 * (100 - CN2)] \qquad \text{Equation 5}$$

SWAT calculates the retention parameter assuming: (1) variable soil profile (2) accumulated plant evapotranspiration. The second method was added to the SWAT 2009 version to account for shallow agricultural soils. The equation for variable soil profile, the retention parameter is calculated according to Equation 6:

$$S = S_{max}\left(1 - \frac{SW}{[SW+\exp(w1-w2*SW)]}\right) \qquad \text{Equation 6}$$

Where $\boldsymbol{S}$ is the retention parameter for a given day(mm), $\boldsymbol{S_{max}}$ is the maximum value that the retention parameter can assume in any given day(mm), $\boldsymbol{SW}$ is the water content in the soil profile excluding the amount of water at wilting point(mm). $\boldsymbol{w1}$ and $\boldsymbol{w2}$ are shape coefficients, computed assuming:

1.Retention parameter for CN1 soil moisture condition corresponds to wilting point of soil profile

2. Retention parameter for CN3 soil moisture condition corresponds to field capacity of soil profile

3. Soil has CN number of 99 when completely saturated

Given conditions stated above, $w1$ and $w2$ are calculated by Equations 7 and 8:

$$w1 = \ln\left[\frac{FC}{1- S_3 * S_{max}^{-1}} - FC\right] + w2 * FC \qquad \text{Equation 7}$$

$$w2 = \frac{(\ln\left[\frac{FC}{1- S_3 * S_{max}^{-1}} - FC\right]) - \ln\left[\frac{SAT}{1- 2.54\, S_{\max}^{-1}} - SAT\right]}{(SAT - FC)} \qquad \text{Equation 8}$$

Where $\boldsymbol{FC}$ is the amount of water in the soil profile at field capacity(mm), $\boldsymbol{S_3}$ is the retention parameter at moist condition for CN3(mm), $\boldsymbol{S_{max}}$ is the retention parameter for moist condition at CN1(mm) and $\boldsymbol{SAT}$ is the amount of water in the soil profile when completely saturated(mm) and 2.54 is the retention parameter at CN = 99. Equations 6, 7 and 8 update CN numbers at each timestep.

Groundwater balances are calculated in the shallow aquifer using equation 9:

$$aq_{sh,i} = aq_{sh,i-1} + w_{rchrg,sh} - Q_{gw} - w_{revap} - w_{pump,sh}$$

Equation 9

Where $\boldsymbol{aq_{sh,i}}$ is the amount of water stored in the shallow aquifer in day i(mm), $\boldsymbol{aq_{sh,i-1}}$is the amount of water stored in the shallow aquifer in day $i-1$ (mm), $\boldsymbol{w_{rchrg,sh}}$is the amount of recharge entering the shallow aquifer in day i, $\boldsymbol{Q_{gw}}$ is the groundwater flow, or base flow into the main channel on day i, $\boldsymbol{w_{revap}}$ is the amount of water moving into the soil zone in response to water deficiencies on day i and $\boldsymbol{w_{pump,sh}}$ is the amount of water removed from the shallow aquifer by pumping in day i (mm).

SWAT computes water percolation in the vadose zone before entering shallow and deep aquifers. The recharge to both aquifers on a given day is calculated by Equation 10:

$$w_{rchrg,i} = \left(1 - \exp\left[-\frac{1}{\delta_{gw}}\right]\right) w_{seep} + \exp\left[-\frac{1}{\delta_{gw}}\right] w_{rchrg,i-1} \qquad \text{Equation 10}$$

Where $\boldsymbol{w_{rchrg,i}}$ is the amount of recharge entering the aquifers on day i (mm), $\boldsymbol{\delta_{gw}}$ is the delay time or drainage time of the overlying geologic formation(days), $\boldsymbol{w_{seep}}$ is the total amount of water exiting the bottom of the soil profile on day i, $\boldsymbol{w_{rchrg,i-1}}$ is the amount of recharge entering the aquifers on day $i-1$. The total amount of water exiting the bottom of the soil profile on day i is calculated: by Equation 11, below:

$$w_{seep} = w_{perc,ly=n} + w_{crk,btm} \qquad \text{Equation 11}$$

Where $\boldsymbol{w_{seep}}$ is the total amount of water exiting the bottom of the soil profile on day i(mm), $\boldsymbol{w_{perc,ly=n}}$ **is** the amount of water percolating out of the lowest layer, n, in the soil profile on day n(mm) and $w_{crk,btm}$ is amount of water flow past the lower boundary of the soil profile due to bypass flow on day i(mm).

Concerning the partitioning between shallow and deep aquifer, deep aquifer percolation is considered a percentage of the water entering both aquifers, as stated by Equation 12:

$$w_{deep} = \beta_{deep} * w_{rchrg} \qquad \text{Equation 12}$$

Where $\boldsymbol{w_{deep}}$ is the amount of water entering the deep aquifer(mm), $\boldsymbol{\beta_{deep}}$ is the aquifer percolation coefficient and $\boldsymbol{w_{rchrg}}$ is the amount of recharge entering both aquifers on the day i(mm).

The amount of recharge entering the shallow aquifer is calculated by Equation 13:

$$w_{rchrg,sh} = w_{rchrg} - w_{deep} \qquad \text{Equation 13}$$

The shallow aquifer contributes to baseflow in the main channel or reach on subbasin. Baseflow enters the reach only if the quantities of water stores in the shallow aquifer exceeds a certain value, user-specified, $aq_{shthr,q}$. The contribution of groundwater flow to reach flow is calculated by Equation 14:

$$Q_{gw} = 8000 * \frac{K_{sat}}{L_{gw}^2} * h_{wtbl} \qquad \text{Equation 14}$$

Where $\boldsymbol{Q_{gw}}$ is the groundwater flow, or baseflow into the main channel(mm), $\boldsymbol{K_{sat}}$ is the aquifer hydraulic conductivity(mm/day), $\boldsymbol{L_{gw}}$ is the distance from the ridge or subbasin divide for the groundwater system to the main channel and $\boldsymbol{h_{wtbl}}$ is the water table height.

On each day, the groundwater flow is calculated according to Equation 15:

$$Q_{gw,i} = Q_{gw,i-1} * \exp[-\alpha_{gw} * \Delta t] + w_{rchrg,sh} * (1 - \exp[-\alpha_{gw} * \Delta t])$$

$$if\ aq_{sh} > aq_{shthr,q} \qquad \text{Equation 15}$$

$$Q_{gw,i} = 0 \qquad if\ aq_{sh} < aq_{shthr,q}$$

Where $\boldsymbol{Q_{gw,i}}$ is the groundwater flow into the main channel on day i, $\boldsymbol{Q_{gw,i-1}}$ is the groundwater flow into the main channel on day $i-1$, $\boldsymbol{\alpha_{gw}}$ is the base flow recession constant, Δt is the timestep, $\boldsymbol{w_{rchrg,sh}}$ is the amount of recharge entering the shallow aquifer on day i(mm), $\boldsymbol{aq_{shthr,q}}$ is the threshold water level in the shallow aquifer for groundwater contribution to the main channel(mm). When the shallow aquifer receives no recharge, $\boldsymbol{Q_{gw,i}}$ is calculated using Equation 16:

$$Q_{gw,i} = Q_{gw,0} * \exp(-\alpha_{gw} * t) \; if \; aq_{sh} > aq_{shthr,q}$$ Equation 16

$$Q_{gw,i} = 0 \qquad if \; aq_{sh} < aq_{shthr,q}$$

More information on the SWAT theoretical can be found on the SWAT theoretical basis manual (NEITSCH et al., 2011) and on the SWAT I/O Documentation (ARNOLD et al., 2013).

3.6 SWAT: Current BMP formulations & equations

Best Management practices modelling in SWAT mainly relies on mechanistical relationships to compute flow and water quality reductions at each timestep. SWAT currently can model several BMPs, at daily and sub daily scales. The currently implemented routines in SWAT are listed below:

(i) Wetlands: Modelling using a water balance for inflow and outflow, as shown in equation 17:

$$V = V_{stored} + V_{flowin} - V_{flowout} + V_{pcp} - V_{evap} - V_{seep}$$ Equation 17

Where $\boldsymbol{V_{stored}}$ (m^3) is the volume of water stored in the wetland at the beginning of day, $\boldsymbol{V_{flowin}}$ (m^3) is the volume of water entering the wetland during the day, $\boldsymbol{V_{pcp}}(m^3)$ is volume of precipitation falling on the wetland during tbhe day, $\boldsymbol{V_{evap}}$ (m^3) is the volume of water removed from the wetland during the day and $\boldsymbol{V_{seep}}(m^3)$ is the volume of water lost from the water body by seepage. Rahman et al. (2016) developed SWATrw, an algorithm that simulates different morphometric formulas and a channel model to reproduce water balances. Applications yield good modelling efficiency values with standard simulations and using(FENG et al., 2013 ;WANG et al., 2010) the Hydrologically Equivalent Wetland concept (WANG et al., 2008).

(ii) Sedimentation-filtration basins:

Jeong et al. (2013) incorporated to the SWAT source code a routine for sedimentation-filtration (SedFil) basin modelling. SedFil basin design is the typical for the Austin (TX) area, which is either partial (Sedimentation and filtration basin hydraulically connected) or full (flow spreading outlet separates chambers and full stormwater volume is to be handled). The module computes saturated and unsaturated flow in sub-daily timesteps, aggregating the results for daily values. Model also computes sediment entrapment.

(iii) Filter strips and grassed waterways:

The Algorithm available in SWAT is a modification of the Vegetative Strip Model (VSMOD), an empirical algorithm developed for computations in sub daily timescales. SWAT brings a modified version, where runoff reduction is calculated by Equation 18, below:

$$R_R = 75.8 - 10.8\, ln * R_L + 25.9 \ln(K_{SAT}) \qquad \text{Equation 18}$$

Where $\boldsymbol{R_R}$ is the runoff reduction in %, $\boldsymbol{R_L}$ is the runoff loading (mm) and $\boldsymbol{K_{SAT}}$ is the saturated hydraulic conductivity (mm). Sediment reductions are based on measured VFS data and is computed using equation 19, below:

$$S_R = 79.0 - 1.04\, S_L + 0.213\, R_R \qquad \text{Equation 19}$$

Where $\boldsymbol{S_R}$ is the sediment is the predicted sediment reduction (%), $\boldsymbol{S_L}$ is the sediment loading(km/m²) and $\boldsymbol{R_R}$ is the runoff reduction (%).

The algorithm also accounts for Nitrogen, Nitrate Nitrogen, Total Phosphorus and Soluble Phosphorus. Grassed Waterways are treated as trapezoidal channels, as width and length are user inputs. Sediment transport capacity is calculated by Equation 20, below:

$$Scap = Spcon * v^{1.5} \qquad \text{Equation 20}$$

Where $\boldsymbol{Scap}$ is the sediment transport capacity(mg/m³), $\boldsymbol{Spcon}$ is sediment transport coefficient and $\boldsymbol{v}$ is the flow velocity in the waterway(m/s). Unsubmerged portions of the waterway act as filter strips trapping both soluble and insoluble pollutants.

Removal of soluble pollutants from the unsubmerged portion is calculated by equation 21, below:

$$Sol_R = 75{,}8 - 10{,}8\log(SD) + 25{,}9\log(SolK) \qquad \text{Equation 21}$$

Where $\boldsymbol{Sol_R}$ is the soluble pollutant removal (%), $\boldsymbol{SD}$ is the runoff depth over unsubmerged waterway area(mm/day) and $\boldsymbol{SolK}$ is the soil saturated hydraulic conductivity(mm/hr.). Removal of particulate pollutant and sediment in the submerged area is calculated by Equation 22, below:

$$Sed_R = 79.0 - 1.04(SedL) + 0.213(Sol_R) \qquad \text{Equation 22}$$

Where $\boldsymbol{Sed_R}$ is the sediment and particulate pollutant removal (%) and $\boldsymbol{SedL}$ is the sediment load per unit area of unsubmerged waterway (Kg/ha/day)?

(iv) Reservoirs:

Reservoir impoundments are located within a subbasin and drain the complete area upstream to it. Equation 23, below, describes the water balance in a reservoir in SWAT.

$$V = V_{stored} + V_{flowin} - V_{flowout} + V_{pcp} - V_{evap} + V_{seep} \qquad \text{Equation 23}$$

Where $\boldsymbol{V}$ is the volume of water in the impoundment at the end of the day(m^3), $\boldsymbol{V_{stored}}$ is the volume of water stored in the water body at the beginning of the day(m^3), $\boldsymbol{V_{flowin}}$ is the volume of water entering the water body during the day, $\boldsymbol{V_{flowout}}$ is the volume of water flowing out of the water body during the day, $\boldsymbol{V_{pcp}}$ is the volume of precipitation falling on the water body during the day(m^3), $\boldsymbol{V_{evap}}$ is the volume of water removed from the water body due to evaporation on the day(m^3) and $\boldsymbol{V_{seep}}$ is the volume of water lost from the water body by seepage(m^3). Empirical equations are used for Surface area, daily rainfall, evaporation and seepage calculations. Outflow can be either supplied by the user or estimated for uncontrolled and controlled reservoirs

Estimated outflow for uncontrolled reservoirs is calculated based on the Principal Spillway volume and release rates and Emergency Spillway volume and release rates. As for controlled reservoirs, a target volume and the timestep required are used as inputs.

3.7 SWAT: BMP Applications – review and opportunities

SWAT equations are suitable mainly for large watershed hydrology. However, adaptations for sub-daily water balances are reported in the literature. Jeong et al. (2010) developed a sub-daily routine for SWAT and calibrated for a 1-year period, 15-minute timestep simulation for the Lost Creek, a 1.9 Km^2 watershed in Austin, Texas with a high Efficiency (R2 = 0.93). The study used the Green and Ampt Mein Larson(GAML) method(MEIN, LARSON ,1973). Modelling efficiency was enhanced due to the physical nature of the GAML hydrograph method, reproducing adequately the watershed quick responses.

The CN method has been proven to have limitations for accounting for surface runoff, mainly for assuming overland flow as stream runoff and underestimating runoff when compared to the GAML method(HUNT et al. ,2009 ; GAREN; MOORE, 2005). The Curve Number provides an estimate to storm event precipitation to direct runoff depth. CN values are preferable to be measured, as tabulated values are estimates for certain soils and under certain conditions (ELI; LAMONT,2010). Although some authors find quite prohibitive the use of CN number for hydrological assessment for more than the original formulation, modifications have been performed in the CN number do account for changing moisture conditions. Furthermore, Brevnova, Eli (2015) conducted experiments comparing Green-Ampt and CN curve numbers. The study concluded that the largest difference in infiltration loss rates occur at low CN values, with less difference at high CN values. Eli; Lamont(2010). compared Green-Ampt and CN-curve numbers and concluded that the deviation between methods is larger for small CN numbers. Viji et al. (2015) compared Green-Ampt and CN, using HEC-HMS simulations. The Curve Number method showed lower performance during peak flow. Given the stated above, the CN number is a feasible tool for runoff estimation, being less effective with larger precipitation events.

Currently, SWAT has BMP routines for (i) ponds, (ii) wetlands, (iii) vegetated buffers, and (iv) agricultural practices: contouring, tillage practices, pesticide and fertilizer applications. SWAT ultimately lumps to the subbasin level all water balance interactions at the HRU level to compute stream flow. Runoff transmission between landscape components is not accounted for. However, the field-scale HRUs representation using SWAT is possible and has been reported in the literature.

GITAU et al. (2008) conducted a BMP evaluation study at field scale in the Cannonsville Watershed Reservoir, using a 1,63 Km², using a BMP equipped and monitored pilot farm and SWAT to evaluate BMP enhancement of barnyard management, crop rotation, strip cropping and tile drains. In this study HRU lumping threshold is set to zero, and field-scale conditions showed diverse efficiency values (NSE ranging from 0.19 to 0.83 for various timesteps). DAI et al. (2017) developed a Fuzzy-credibility Constrain Algorithm (FCCP) for SWAT in a 300 Km² watershed in China. In this study, the HRU parameter reparameterization combined with pre-existing SWAT routines to simulate the effect of Parallel Terracing (SLSUBBSN, USLE P, CN2), Grade stabilization (CH_S2, CH_EROD) Filter, Strips (FILTERW) and Detention Wetlands (WET_NVOL, WET_FR, WET_NSA). Sheshukov et al. (2016) tested a 897 Km² Watershed for cattle pasture fencing and off-stream watering. Daggupati et al. (2010) tested BMP and tile drainage for the 489 Km² Little vermilion river.

SWAT also has been target of research on coupling urban BMPs and LID structure algorithms. Jeong et al. (2013) developed a sedimentation-filtration (SedFil) algorithm for SWAT. Two types of SedFil basins are available: partial and full, with sizing and constructive design typical for the Austin, TX area. The algorithm requires sub-daily data and accounts for both saturated and unsaturated fluxes. Shannak (2017) modified SedFil routines to perform as permeable pavement and rain gardens an obtained good result testing for a small watershed in in Austin, TX. More recently, HER et al. (2017) developed a distributed HRU algorithm for hydrologically accounting for spatially distributed BMPs, treating each LID as isolated or connected tanks, each accounting for infiltration, discharge and evaporation processes at the sub-daily scale. The algorithm simulated green roofs, rain gardens, cisterns and porous pavements at the 1.49 Km² Brentwood watershed in Austin, TX. SWAT was run using high resolution data (0.6 m resolution DEM and LULC maps) and simulated rainfall. Results were satisfactory both computationally and field-scale wise.

Despite the numerous studies concerning SWAT and BMP applications, limitations and uncertainty in BMP modelling always must be taken into account. Hunt et al. (2009) discusses limitations and performances of BMP devices reporting that BMP performances do not remain steady during device life cycle and may be pollutant sources under certain antecedent dry conditions and under specific operation and management activities. In addition, the water balances in SWAT are still not sensitive to individual landscape mass exchanges. Models that incorporate such features are more likely to describe the system in a more physically-based fashion, with the disadvantage of requiring significantly larger datasets and computational times.

On his paper, Elliott; Trowsdale (2007) reviews several LID modelling applications, and points out that not enough constituents are represented within these models. Ahiablame et al. (2012) on an literature review article on LID effectiveness, points out that not many studies have been performed to evaluate urban BMP effects at the watershed scale, and highlights that field-scale analysis is practically impossible to expand to watershed scale and that there are not still enough BMP water quality studies to completely comprehend BMP influences at larger scales. The latter two conclusions are shared by Liu et al. (2017) and Hunt et al. (2009). The latter also states that, nor studies are enough in number or in monitoring length. Another discussion point in the literature is that mainly in the studies validating LID and BMP devices, no design specifics are stated, leaving a gap in reproducibility and study transferability(LIU et al., 2015).

HUNT et al. (2009) also points out that SWAT still faces challenges in BMP modelling as its current theoretical assumptions and computational capabilities are still in development:

(i) Does not consider landscape water transmission

Considers Wetlands and ponds as new Subbasins, not allowing HRU-scale impoundments to be simulated.

(ii) Considers reservoirs as Subbasins, not allowing HRU individual impoundments to be reproduced

(iii) The CN method, despite requiring less data, is not a recommended model for BMP simulation.

*Simulations may be enhanced by models that account for changing moisture conditions

(iv) Lacks in number of constituents modeled for Urban BMPs

A compilation of studies of Best Management practices application in SWAT is Summarized in Table 3, next.

Table 3 - Compilation of SWAT and BMP studies

Reference	Location	Watershed name	Watershed Area	Timestep	BMPS represented	Observation
Ozcan (2012)	Ankara, Turkey	Lake Mogan	970 Km^2	Daily	Fertilizer reduction No tillage Contouring Terracing	Arid region/few data
Gitau (2008)	Delhi County, NY	Cannonsville reservoir	1.63 Km^2	Daily (weather gen)	Barnyard managemt Crop rotation Strip crop Tile drainage	Field-scale validation w/ pilot farm
Jeong (2006)	Austin, TX	Jollyville sand filter	0.0281 Km^2	15 min	Sedimentation-filtration Basins	Typical Austin design
Shannak (2017)	Austin, TX	Blunn Crekk	2.56 Km^2	15 min	Rain gardens Permeable pavement AS SedFil	Sediment delivery enhancements
Haas (2017)	Northen Germany	Treene River	481 Km^2	Daily	Buffer Strips Pasture land increase Use of corn for biogas Fertilizer reduction	Economic assessment
Dai (2016)	Kumming, Chuna	Lake Dianchi	300 Km^2	Daily	Terracing Grade Stabilization Filter Strips Detention Wetlands	Fuzzy credibility-chance algorithm (FCCP)
Her (2016)	Austin, TX	Brentwood	14.98 Km^2	15 min	Green roof Rain gardens Cisterns Porous pavements	High resolution DEM(0.6m) Development of LID module
Motsinger (2016)	Vermillion County, IL	Little Vermilion river	489 Km^2	Daily	Reduced tillage Filter Strips Wetlands Tile drainage Cover Crops	-
Sheshukov (2016)	Eastern Kansas	Pottawatomie Creek	897 Km^2	Daily	Stream Fencing Off-stream pasture	Synthetic off-stream watering & fencing

SWAT is currently developed and distributed by the TEXAS A&M university (TAMU), and the TEXAS AgriLife Research Center. Other packages are developed within the USDA & TAMU. The Agricultural policy Environmental Extender Module is another watershed model developed for agricultural and urban watersheds. The Conservation practice modelling guide for SWAT and APEX provides resources for agricultural BMP modelling, using a surrogate approach, consisting in re-parameterizing HRUs and using algorithms already present in SWAT

Despite all limitations to diffuse pollution modelling, BMP modelling and BMP modelling in SWAT, lumped modelling has been performed before. AHIABLAME et al. (2012) modelled successfully an 426 dwelling sub-urban area in Lafayette, NC, investigating BMP water quality and quantity enhancements using seven BMPs (bioretention, open-wooded spaces, green roofs, rain barrels/cisterns, porous pavements and permeable patios) using the Long-Term Hydrologic Impact Assessment–Low (L-THIA) and CN lumped approach and as boundary condition to BMP is sized to remove a proportional value to BMP area itself.

3.8 Key points

SWAT has been extensively validated and researched. BMP representation could be divided in two branches: detailed and lumped approaches.

About urban BMPS in SWAT, main research fields are: (i) optimization of BMP placement and sizing, (ii) BMP representation capability enhancement and (iii) SWAT adaptation for mode detailed LID representation are developing fields. An adequate balance between detailed representation, data needs, operational complexity and the existence of long-term data on BMP performance for various constituents is still an issue for SWAT and other models.

Large watershed hydrology cannot be particularized to field-scale due to simplicity of approach. The contrary is too costly and potentially careless. SWAT is a model majorly applied to agricultural watersheds, having successful applications for agricultural and rural practices, climate and LULC changes.

4 MATERIALS AND METHODS

4.1 Hypothesis & Simplifications

The Barigui River basin suggests potential flood control and diffuse pollution measures, as stated in Section 1 (Introduction). The mostly used open-source watershed simulation models account for field scale mitigation strategies on sub-daily timesteps and/or requires massive amounts of data (SHOEMAKER et al., 2005). To comprehend the watershed scale effects of inserting BMPs, the approach of this work was to assess both on a field-scale timesteps up to annual averages, in an attempt to simulate both the new geographical and infiltration conditions.

The choice of SWAT was due to its land and channel phase detailed representations. SWAT soil reservoir models are dependent on HRU Groundwater Flow slopes and Groundwater reservoir slopes (HRU_SLP and SLSOIL, respectively), in addition to drainage length for subsurface flow (SLSUBBSN). Hence, it allows for special geographical representation at the HRU scale, as different HRUs are created and their relationship is not linear. As SWAT has a powerful integration with GIS, the capability of simulating hydrological changes is significantly facilitated. Also, the opportunity of creating code for faster SWAT parameterization and GIS BMP scenario generation may be a promising field concerning the development of such models and their associated applications.

In this study case particularly, high resolution LULC, parcel and street layout datasets subsidize the BMP distribution and model input. Despite the limitations inherent to hydrological/pollutant fate and transport models, as in Section 3.3, SWAT is assumed to adequately represent hydrology and water quality processes in watersheds, with feasible application to the study case, recognizing that this work focuses on the main hydrological changes.

To simulate BMP effects, the hypothesis of an HRU working essentially as a recharge unit is employed. The parameter that controls water input to soil and in a larger timescale, to all reach contributions on SWAT is the Curve Number. Hence, a recharge unit in this work is defined as a CN reduced area on a feasible location within the watershed.

It is not the goal of this study to determine in a timely-responsive fashion BMP implementation effects on hydrographs, neither to quantify BMP effects as isolated devices, but to lump BMP effects at the subbasin level and evaluate the effects of such devices at daily scales on water quality and quantity, aiming to assess their variation in time and space as recharge zones are created, in an attempt to observe what changes in a watershed are possible to be seen with the BMP implementation. Figure 7 shows a simplified study scheme, synthesizing the hypothesis, simplifications, challenges and expected outcomes of this work.

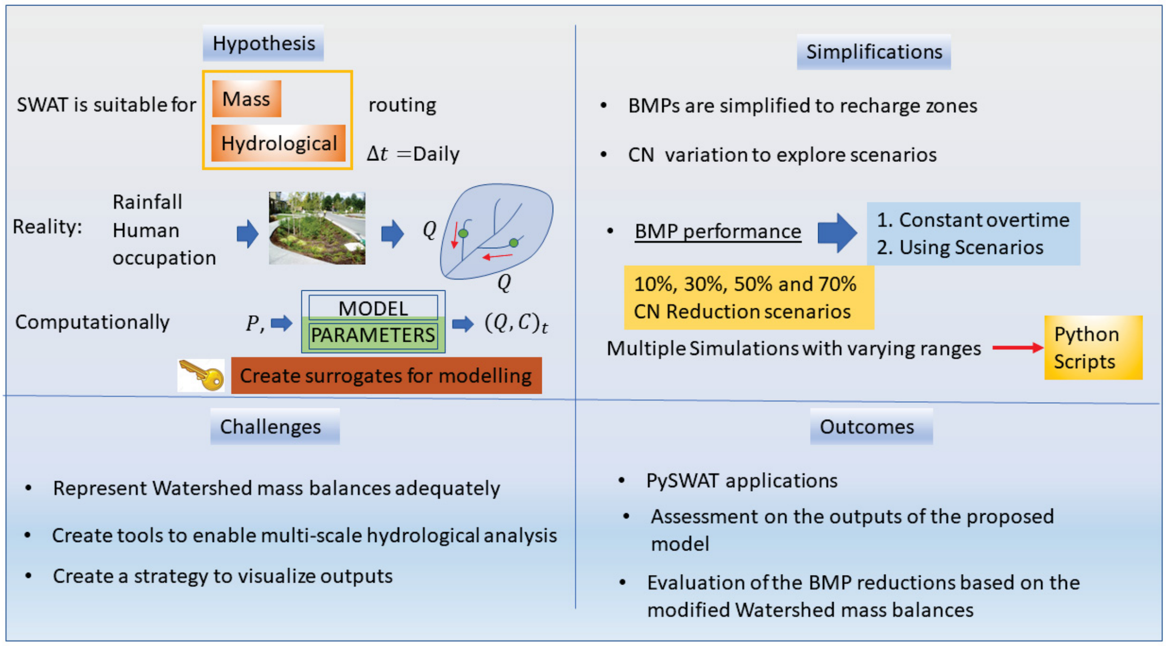


Figure 7 - Study scheme

This "computational BMP" scheme aims to lump at the subbasin and watershed level from the computing of individual influences of BMPs placed at the field-scale level. The schemes are based in reductions applied to the Curve Number and applied top soil layer amendments. For water quality enhancement, reduction of export concentrations on HRUs are employed. For the implementation of this method, three phases are considered: (i) BMP spatial location, (ii) BMP coupling to SWAT input maps as modified land uses (iii) BMP enhanced areas (BEA) re-parameterization and (iv) testing for uncertainty.

4.2 Study area

The Barigui River basin, depicted in Figure 8 is in the Southwestern portion of the state of Parana, Southern Brazil. It is a tributary of the Iguassu River and it belongs to the Upper Iguassu River basin. The Barigui River basin has an area of 264.89 Km² and a stream length of 63 km. Data sources are showed in Table 5.

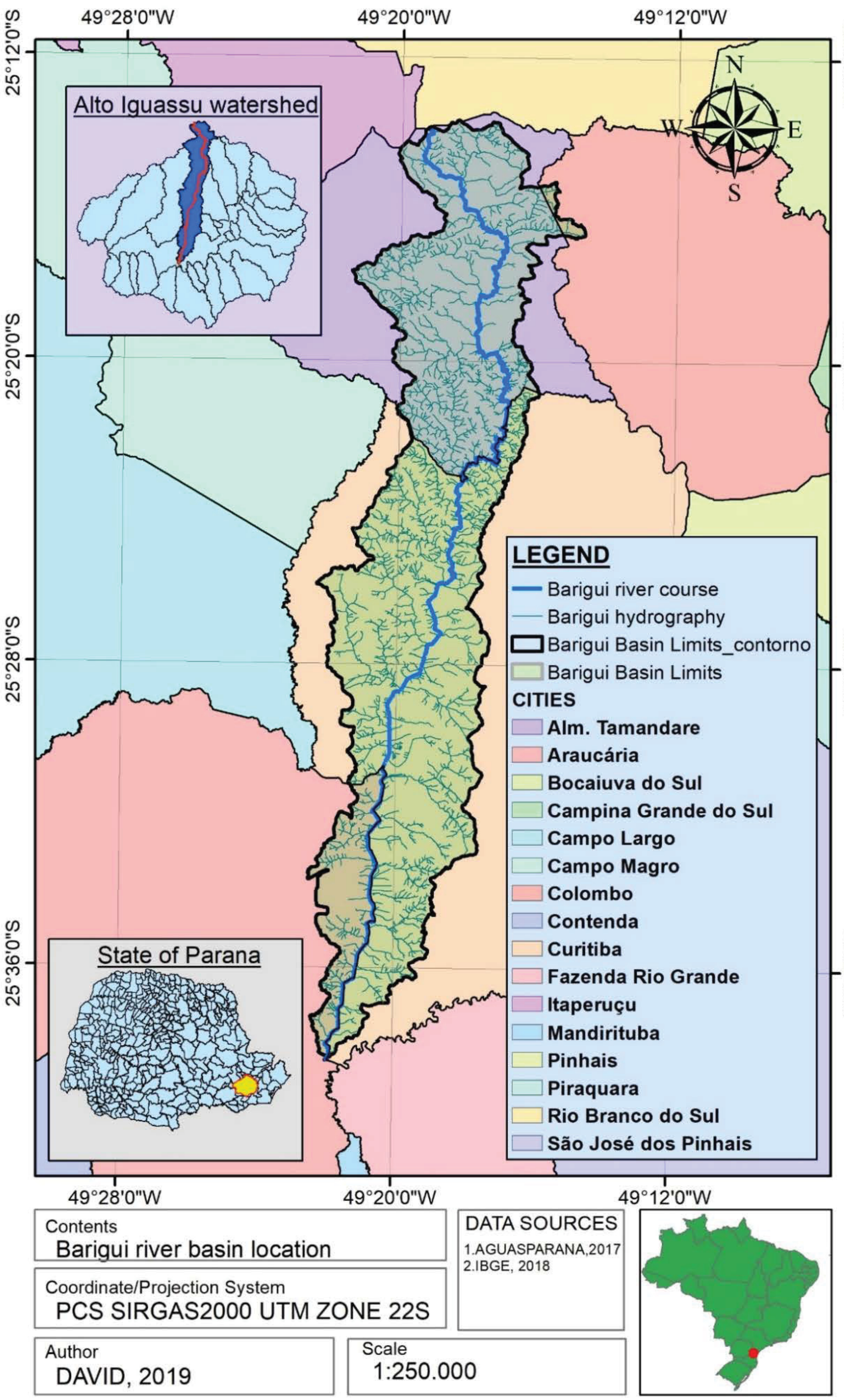


Figure 8 - Barigui river Basin Location

The Barigui basin is distributed along five municipalities. Its spatial distribution within municipalities is described in Table 4.

Table 4 - Municipalities within the Barigui River basin

City	City area (Km²)	Basin area on city (Km²)	Percentage of the Town/City on the Barigui Basin
Araucaria	469.94	23.10	4.91%
Almirante Tamandare	193.97	96.02	49.50%
Curitiba	434.55	144.14	33.16%
Colombo	197.16	1.608	0.81%
Rio Branco do Sul	819.66	0.012	0.0014%
Total		264.88	

Source: FERREIRA; FERNANDES (2016)

The most predominant land uses in the basin are fields, natural arboreal vegetation and medium density urban areas. Figure 9 shows the land use distribution upstream to the outlet point. Figure 10 shows the reclassified LULC for the Barigui Basin. The 21-original dataset LULCs were reclassified from the feature referenced in Table 5, and reclassified into 5 categories according to the table in Annex 3 – Section 8. SWAT model was run using the reclassified map, as the originally estimated CN and other LULC-dependent parameters did not deviate among LULCs. Also, the difference between the 21 and 5 LULC was negligible and computational times increased significantly, as shown in the preliminary model runs.

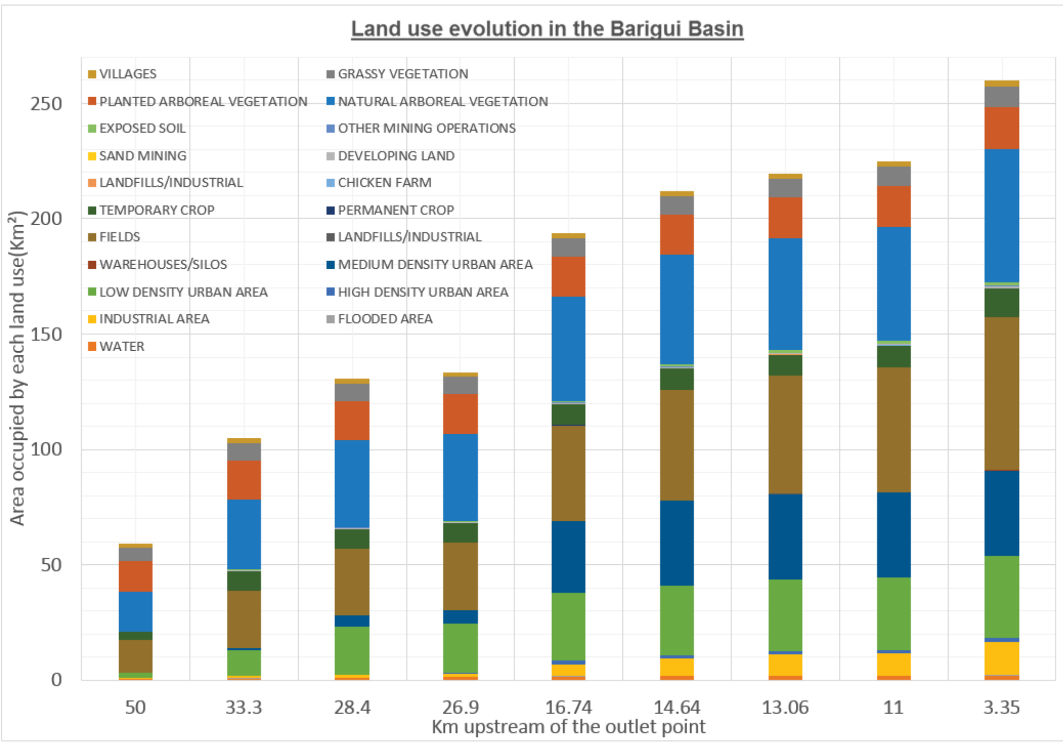


Figure 9- Land use evolution upstream to outlet point of the Barigui Basin

Source: FERREIRA; FERNANDES (2016)

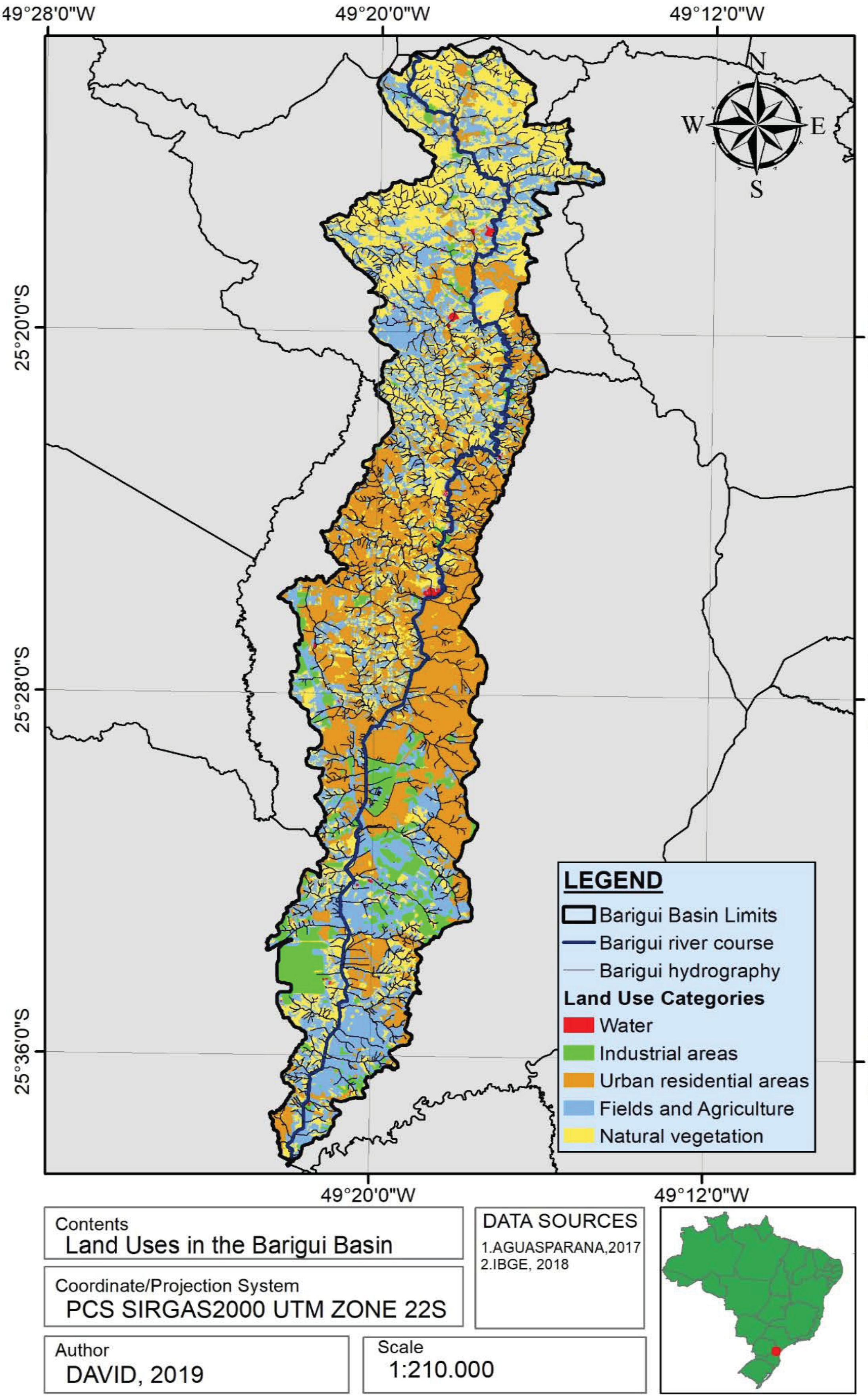


Figure 10 - Barigui Basin LULC map

The most dominant land uses, according to the Annex 1 Table and Figure 10 above, are Fields, and Arboreal Planted vegetation. However, medium and low urban areas occupy significant portions of land within the watershed, which grants a mixed LULC characteristic to the basin.

4.3 Geospatial data sources

The Land Use and Land Cover (LULC), and a 10-m resolution DEM obtained from the Aguas Parana Institute will be used as inputs for HRU delimitation Soil data was obtained from the Instituto Paranaense de Assistência Técnica e Extensão Rural (EMATER). Spatial data sources are shown in

Table 5.

Table 5 - Spatial data sources

Content	Type	Source
LULC	Shapefile	AGUASPARANA (2017)
Soil Map	Shapefile	EMATER (2007)
Digital elevation model (DEM)	Raster	AGUASPARANA (2017)
Reservoir map	Shapefile	IPPUC (2018)
Stream grid	Shapefile	AGUASPARANA (2017)
Groundwater Depth (Well static levels)	Shapefile	AGUASPARANA (2018)

All geospatial data is projected in UTM zone 22S coordinates in the SIRGAS 2000 datum. All data used in this study is public.

4.4 Precipitation and weather gaging Stations

A search on the Hidroweb (ANA, 2017) database showed a total of 19 available stations, with various initial and final monitoring periods. Missing data periods were analyzed to assess the data quality and availability of rainfall gaging stations. Six stations were found eligible for modelling in SWAT (i.e.: presented continuous rainfall registers within the modelled period). Two out of the six stations are in the Barigui watershed.

Table 6 synthesizes the stations used in this work.

Table 6 – Hidroweb station codes and available periods used for this study

Name	Hidroweb code	Available period	Missing data
Areias	02549107	2000-2017	None
Almirante Tamandare	02549100	2000-2017	None
Colonia D. Pedro	02549080	2000-2017	None
Juruqui Landfill	02549077	2000-2017	01/Jan/14-22/set/15
Sanepar intake at Passaúna reservoir	02549081	2000-2017	01/Jun/10-16/set/10
Simepar	02549101	2000-2017	None

The filling for missing periods was performed using geospatial interpolation. It was performed using the Kriging method, based on the spherical Semi variogram method, using data from the nearby stations. The method was implemented using the ArcGIS Geostatistical Analyst (ESRI, 2015) platform. Figure 11, below, shows precipitation, gaging and climate stations used in this study.

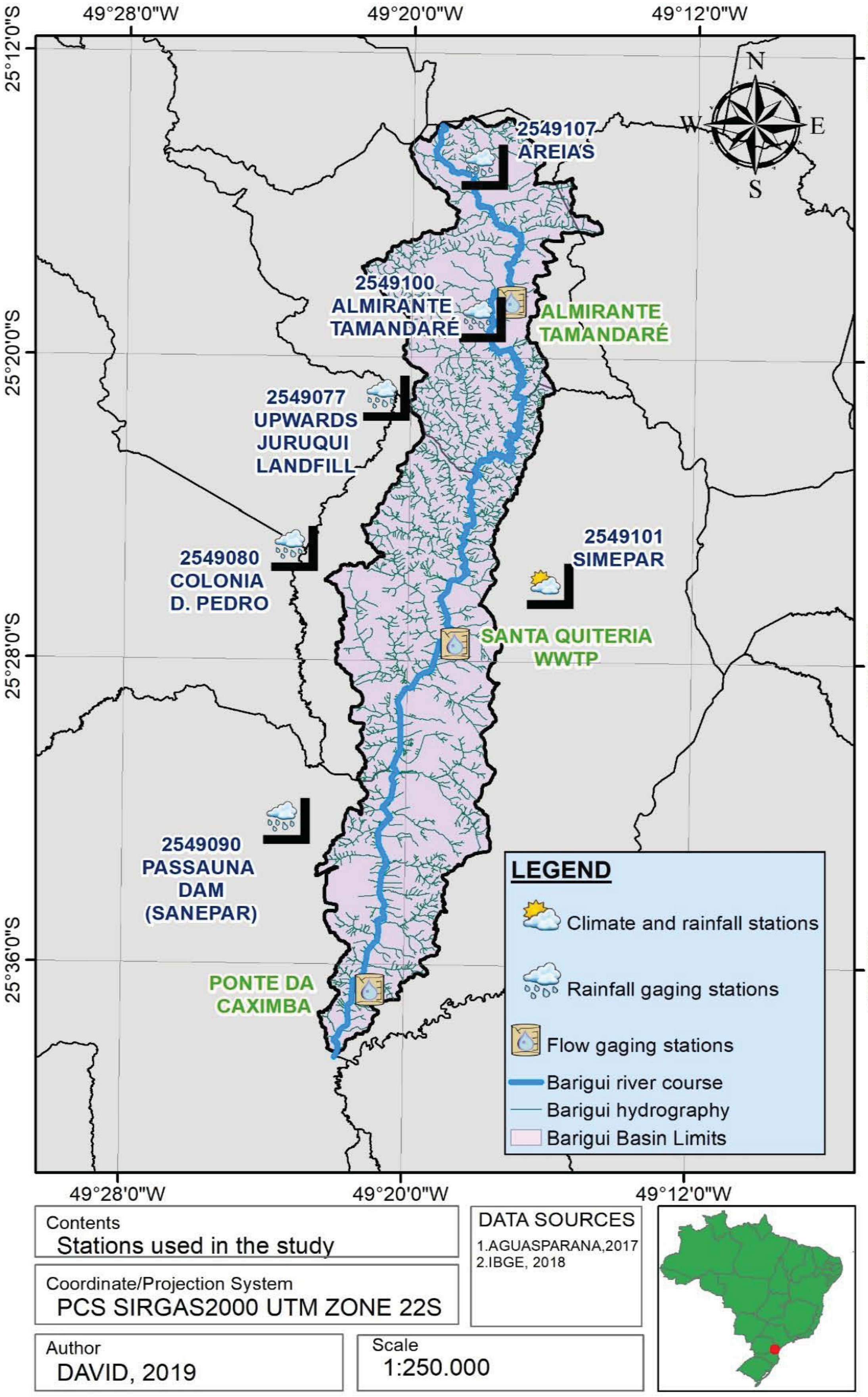


Figure 11 - Rainfall and weather stations in Barigui river basin

As for weather (i.e.: other parameters than rainfall), the SIMEPAR (SIMEPAR. 2017) station data was used. The station measures several parameters every 15 minutes. Daily average solar radiation, relative humidity, maximum and minimum temperatures, wind speed and accumulated daily rainfall were obtained from the station. Data consistency and daily average transformations were performed using two Python packages: Pandas and Numerical Python (NumPy). The six rainfall gaging stations as well as the weather data station location are shown in Figure 11.

4.5 Flow gaging data

Flow gaging in two stations was employed for flow calibration: (1) Santa Quiteria (SQ) and (2) Ponte da Caximba (PC). The points are depicted in Figure 11. Within the city of Almirante Tamandare, as depicted in Figure 11, is located the Sao Jorge Wastewater Treatment Plant (SJWWTP). Its daily outflows were obtained from the geospatial permit database from the AGUASPARANA institute (AGUASPARANA, 2017), with an average value of 0.147 m^3s^{-1}.

The SQ station is located upstream to the Santa Quiteria Wastewater Treatment Plant, 22.934 km upstream the basin outlet. It drains a total area of 152.65 km², with main land uses upstream to point: Natural Arboreal Vegetation (41.11 km²), Fields (31,92 km²), Low Urban area (25,15 km²) and Planted Arboreal Vegetation (17,15 km²). Flow monitoring is available between Nov/2004 to Feb/2017. Major non-observed values were in 2005 and 2012, as shown in Figure 12. The station is operated by Aguas Parana Institute, under the Hidroweb code 65019675. Daily flows from the Santa Quiteria WWTP (SQWWTP) were obtained from SANEPAR (SANEPAR, 2017), with an average value of 0.280 m^3s^{-1}.

The CX station is located 3.758 km upstream to the outlet point, in the southern part of the basin. It drains an area of 246.53 km², with main land uses upstream to point: (i) Fields (60.52 km²), (ii) Natural Arboreal vegetation (54.04 km²), (iii) Medium Urban Areas (36.69 km²) and (iv) Low Urban Areas (35.08 km²). Flow registers are available from 2000 to Jan/2015, as shown in Figure 12.

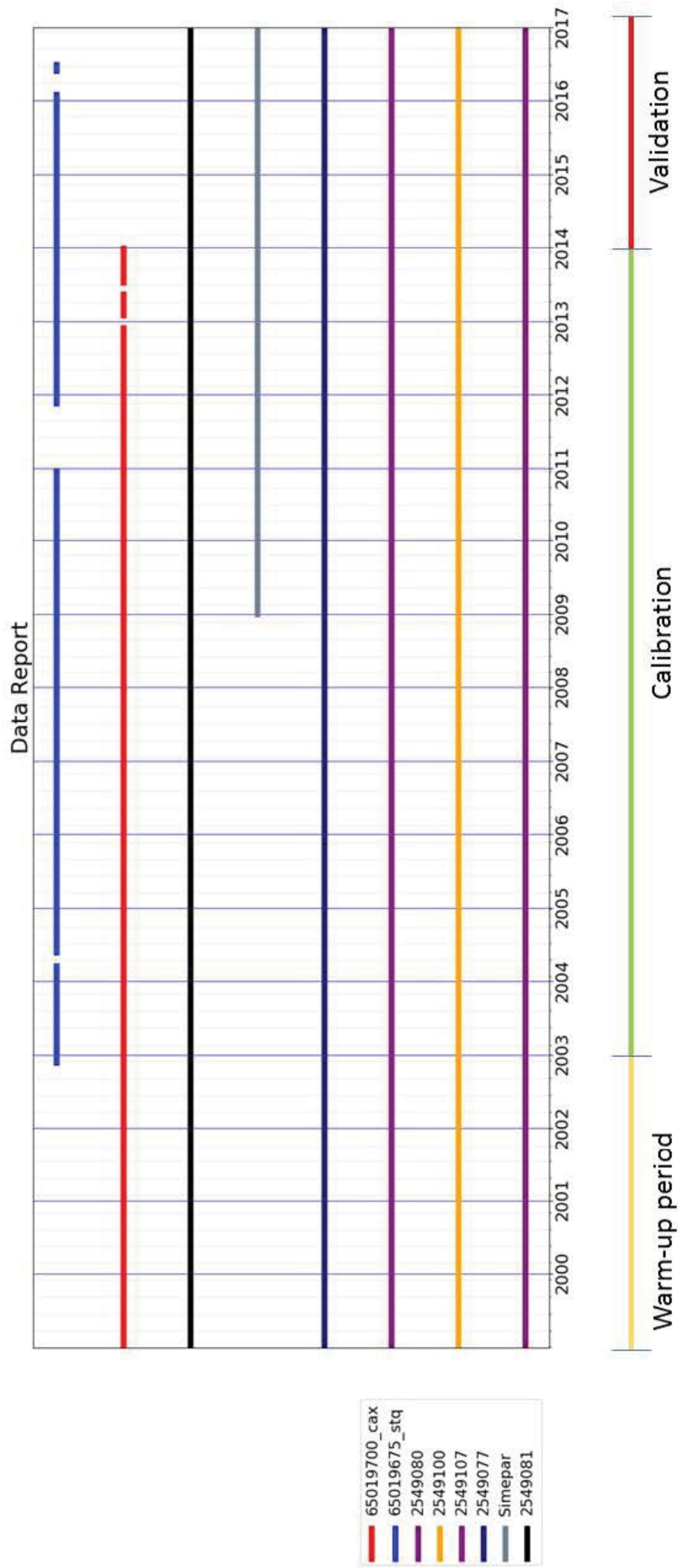


Figure 12 - Available data periods

Missing data periods are present between Dez. /13-Jan/14 and Jun./2014. The station is operated by the Aguas Parana Institute, and monitoring has been discontinued in Feb./2015. The calibration and validation period choice were motivated due to optimize the time-series efficiency, exploring at most the available data.

4.6 Soil Data

Soil data was obtained as geospatial features. The dataset shows that for main types of soils are present in the study region.

Table 7 – Soil layer properties for the Barigui Basin

Soil type	Number of Layers	Hydro logical Group	Layer depth (mm)	Density (g/cm³)	Ksat (mm/hr)	%Clay	%Silt	%Sand	Albedo (%)	USLE K
Cambisoil	2	C	200	0.81	16.56	52	25	23	0.15	0.0145
			800	0.86	14.09	53	26	21		0.0193
Gleisoil	2	D	300	1.66	1.63	45	32	23	0.05	0.049
			600	1.66	1.08	48	32	20		0.049
Latosoil	8	A	240	0.71	6.45	85	12	3	0.1	0.0187
			560	0.8	3.95	87	11	2		0.0201
			1010	0.9	2.02	87	11	2		0.0205
			1300	0.86	3.92	89	9	2		0.0192
			1950	0.84	5.27	89	9	2		0.0227
			2500	0.84	5.27	90	8	2		0.0229
			2920	0.84	5.27	89	9	2		0.0247
			3340	0.84	5.28	85	13	2		0.028
Argisoil	6	C	130	1.42	36.4	12	29	59	0.1	0.0211
			360	1.39	26	18	28	54		0.0223
			670	1.54	1.1	35	23	42		0.0199
			1100	1.67	0.07	40	23	37		0.0194
			1480	1.67	0.24	34	26	40		0.0206
			1900	1.72	0.11	34	28	38		0.0209
Urban	1	C	200	1.5	500	15	30	55	0.23	0.005

Source: MUHLENHOFF; FERNANDES (2012)

4.7 Conceptual model

A conceptual model for a SWAT model must determine what are the conditions to be reproduced, taking into consideration data limitations and quality, and the limitations that are acceptable for a certain model. Three schemes were attempted for this model. The first did not included point sources or reservoirs in the watershed. A second model with the Parque Barigui and Tingui reservoirs and the three main point sources was built. However, the simulations did not reproduce the Barigui or Tingui reservoir adequately and hence were removed. A third conceptual model involved the three WWTPs on the stream course (Sao Jorge, Santa Quiteria and CIC Xisto) and other 10 large point source contributors to the river, as depicted in Figure 13.

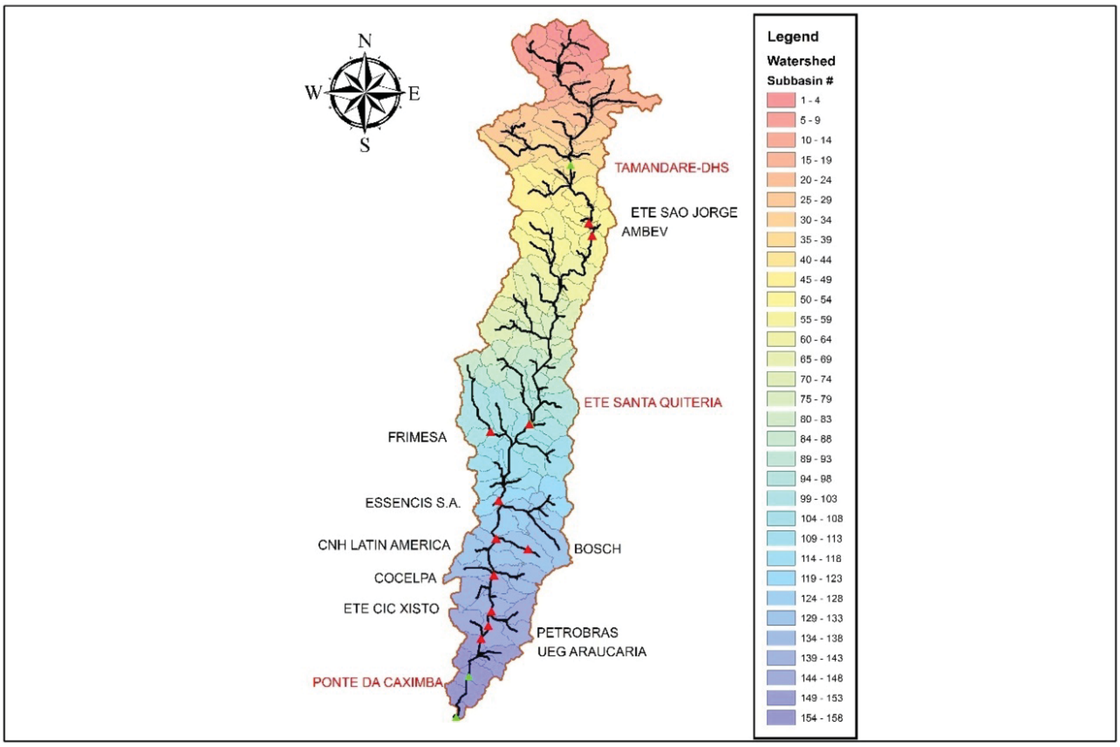


Figure 13 – Conceptual model for diffuse pollution in the Barigui River basin

Point source data(Table 8) was obtained from the AGUASPARANA institute effluent licensing program (AGUASPARANA, 2017), and daily permit values were inserted as point sources in SWAT, a feasible approximation since the point sources are major industries and WWTPs. The effect of reservoirs was neglected in this third and final conceptual model, and point sources contributed significantly to an increase in simulation efficiency.

Table 8 - Point source inputs for the Barigui River basin SWAT Model

User	Activity	Subbasin #	Effluent flow(m^3/h)	Effluent flow (10^4 m^3/day)
Sanepar (WWTP Sao Jorge)	WWTP	52	530	12720
Ambev	Beverage production	57	60	1440
Sanepar (WWTP Santa Quiteria)	WWTP	99	1008	24192
Frimesa	Dairy production	103	26.9	645
Essencis LTDA	Environmental cleanup	122	35.6	855
CNH Latin America	Civil construction/truck vehicle manufacture	128	59.3	1422
Bosch	Electical parts	132	6.7	160
COCELPA	Paper mill	139	354.2	8500
Sanepar (WWTP CIC Xisto)	WWTP	143	1291.7	31000
Petrobras	Oil Refining	148	75	1800
UEG Araucaria	Themal Power plant	149	116.7	2800

The resulting model was constructed with a 100-ha limit for river delineation, using built-in ArcSWAT ArcHydro module. An HRU delineation was set to 0% threshold, 2401 HRU objects and 155 sub-basins were generated.

4.8 Flow calibration

4.8.1 SWAT Calibration and Uncertainty Program (SWAT – CUP)

The SWAT Calibration and Uncertainty Program (SWAT-CUP) (ABBASPOUR ,2015) assists calibration on pre-defined parameter ranges. The program has four routines: Sequential Uncertainty Fitting (SUFI2), General Likelihood Uncertainty Estimation (GLUE), Parameter Solution (PARASOL) e Particle Swarm optimization (PSO) for uncertainty estimation and parameter adjustment. SUFI2 has been successfully used in the literature (ABDELWAHAB et al. ,2018;MOLINA-NAVARRO et al. 2017;PATIL; RAMSANKARAN, 2017 ; PEREZ-VALDIVIA et al.. 2017 MOTSINGER et al., 2016; BRIGHENTI et al., 2016), being a powerful, well documented and short implementation time tool for parameter calibration.

SUFI2 uses Latin Hypercube for multiple parameter sampling within established ranges. For each sample, a simulation is performed, and efficiency and uncertainty parameters (t-stat and p-value) are computed. Each simulation also computes an efficiency indicator. Currently nine objective functions are available. The Nash-Sutcliffe Efficiency (Nash; Sutcliffe (1970) is an indicator of goodness of fit widely used and recommended as flow discharge objective function (ARNOLD et al., 2012; D. N. MORIASI et al., 2007). Equation 25, below, depicts the Nash-Sutcliffe Efficiency indicator:

$$NSE = 1 - \sum_{t=1}^{T} \frac{(Q_{modelled,t} - Q_{observed,t})^2}{(Q_{observed,t} - \bar{Q}_{mean,obs})^2} \qquad \text{Equation 25}$$

Where $\boldsymbol{Q_{modelled,t}}$ is the simulated value at timestep t, $\boldsymbol{Q_{observed,t}}$ is the measured value at timestep t, and $\boldsymbol{\bar{Q}_{mean,obs}}$ is the mean observed value in the period.

Another indicator employed in this work is the R^2 indicator, computed by Equation 26, below.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad , with\ SS_{tot} = \sum_i (\hat{Q}_t - \bar{Q})^2 \ and\ SS_{res} = \sum_i (Q_t - f_t)^2$$

Equation 26

Where $\boldsymbol{SS_{res}}$ is the regression sum of squares, $\boldsymbol{SS_{tot}}$ is the residual sum of squares, $\boldsymbol{\widehat{Q_t}}$ is the simulated value on timestep t, $\boldsymbol{\bar{Q}}$ is the mean observed value, $\boldsymbol{Q_t}$ is the observed value on timestep t.

R^2 was selected as a measure as it shows the simple co-variance-based fit between two functions. Comparing different objective functions provides interesting evidence concerting parameter optimization and model and parameter behavior, as each function trends to assess efficiency differently (ABBASPOUR, 2005).

The calibration strategy consisted in achieving both efficient and physically adequate parameter ranges to represent watershed processes (ABBASPOUR et al. ,2015). Such process might be achieved through manual or automatic calibration. Abbaspour (2005) recommends designating a calibration strategy, with (i) selecting relevant and negligible processes(ii)selecting relevant parameters (iii)define a calibration strategy, based on what are the desired calibration outcomes. SWAT-CUP calibration assists parameter sensitivity, parameter adjust and uncertainty evaluation. Calibration based at the HRU level is feasible in SWAT cup, which allows for heterogeneous spatial description (ABBASPOUR et al., 2017). In situ watershed knowledge is also desirable and can be a very important asset to a realistic calibration scheme (PANAGOPOULOS et al., 2011) There is more than one approach to calibration-validation process, concerning the division of calibration and validation periods and its data sources. The split-sample method suggests dividing simulation periods in 50% calibration-50% validation. Other procedures of calibration are the Differential split sample, proxy catchment (KLEMEŠ, 1986) and Crash Test subsampling (ANDRÉASSIAN et al., 2009) and others. In this work, the split sample calibration is used. According to KLEMEŠ (1986), is not necessary that calibration and validation periods must be equally split, depending on goals, data availability and calibration goals.

4.8.2 Calibration procedures

Flow gaging stations in this study have different available data periods, as illustrated in Figure 11. The calibration period was established aiming to account for most of the watershed processes, but also allowing for validation of the TM station. The calibration and validation periods were established between 2006-2014 and 2014-2017, respectively.

The calibration procedure was performed in four steps, following SWAT-CUP manual and other recommended references (ABBASPOUR et al. ,2017, 2015 ; ARNOLD et al., 2012 ; SWAT USER GROUP, 2013). The steps are: (i) Establishing the processes to be represented in the watershed, choosing which ones should be reproduced and which to be considered not relevant or not important to this study, (ii) Performing sensitivity on elected parameters to assess their influence in the model results (iii) Iterations for adequate range fit, (iv) Validation for the defined period. Each of the phases is detailed further.

(i) Establishing calibration hypothesis

The Barigui River basin flow monitoring points are strategic points as they show significant land use changes, according to the described in the study area section. Land uses in the region upstream to TM station are mainly agricultural and vegetation (fields, temporary crops and natural and planted arboreal vegetation). Previous studies (FERREIRA; FERNANDES, 2013) show that groundwater processes severely affect hydrological balances, yielding large water quantities and causing groundwater flow to influence in wet-condition runoff discharge. In-depth description of such phenomena would require more thorough analysis and quantification, being limited to the relationships that SWAT established through the hydrological cycle. Hence, this model is subject to such limitations. Having that in mind, it is established that calibration of basins 1-39 would be mainly dependent on groundwater process and on surface runoff decay, with eventual adjustments for peak flow.

As for the incremental basin between TM and SQ, it shares similar land use distribution with the addition of low urban areas and medium density occupied areas, where soil water and groundwater transport processes contribute less to streamflow. Additionally, non-urban uses within this area also may show larger impervious areas, in addition to uncovered soils. Satellite imagery Analysis suggests also that overland flow, channel roughness and lateral flow inflow rates could significantly affect the distribution of water balances. Hence, calibration of subbasins 39-98 mainly should rely on parameters that reduce groundwater flow, increase peak flows can account for recession properly.

As for the incremental area between SQ and CX, land use dynamics shows intense low urban and industrial areas, in the regions of the neighborhoods of Cidade Industrial de Curitiba, Pinheirinho, Santa Quiteria and Campo de Santana. Satellite imagery analysis, along with previous experience in the basin and hydrograph analysis raises the following hypothesis: (i) lateral flow and groundwater flow processes do not affect the water balance significantly, (ii) High peak flows on hydrographs, which suggests surface runoff is a major source of flow during wet periods, (iii) Overland flow and stormwater delivery speed is higher.

The hypothesis stated above was used to establish eligible parameters for One at a time (OAT). Also, the three region calibration schemes have been tested for baseflow decay factor (ALPHA_BF), both globally and accounting separately for the specific LULCs: VARN (Vegetação Arbórea Natural/Natural Arboreal vegetation), VARP (Vegetacão Arbórea Plantada), CMPO(Campo/Fields), and AUBD (Area Urbana Baixa/Low Urban Area). The hypothesis test aimed to verify if varying these parameters for area wise dominant urban and nonurban land uses could effectively change baseflow in gaging stations.

Within the stated hypothesis above, the strategy to represent these specific processes within the regions TM, TM-SQ and SQ-CX, being each station verified for sensitivity and calibrated separately with the upstream regions calibrated. In practical terms, sub basins 1-39, 40-98 and 99-155 were calibrated separately.

(ii) Performing sensitivity analysis

26 parameters were selected along the three regions for sensitivity analysis. One at a time (OAT) analysis was performed aiming to determine sensitive parameters. The 20 most significative parameters resulting from the All at a time (AAT) analysis were selected for parameter iteration. SWAT CUP distinguishes more, and less sensitive parameters based on p-value and t-stat on for a linear regression-based on objective function. Some parameters deemed too relevant for calibration were also selected for parameter iteration, based on the hypothesis that these parameters combined could eventually reach a degree of significance in the model. Figure 16 shows the input parameters used as inputs for sensitivity analysis.

(iii) Iterating for adequate ranges

Having the adequate parameters known, five 400 iterations were performed in SWAT-CUP to find adequate parameter ranges. Despite the SWAT-CUP manual recommendation for three-to-five 350-500 iterations, p-values and t-stat values converged to zero in these scenarios. New parameter ranges calculated by each iteration were verified for inconsistencies before the next iteration.

(ii) (iv) Model validation

Model validation was conducted using the 2015-2017 period and the adjusted parameters, comparing both time-series for the same efficiency indicators: NSE and R2. A synthesis of calibration and validation processes, with work-packages distributed in time is shown in Figure 14.

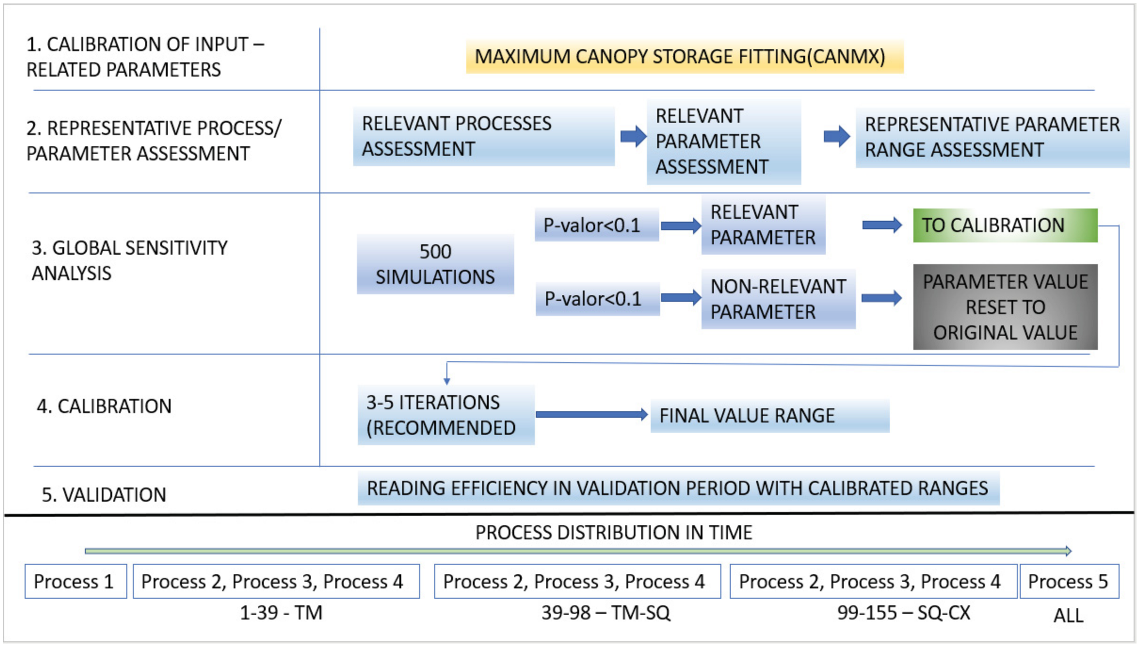


Figure 14 - Calibration scheme

4.8.3 BMP flow reduction computation

The effects of BMP implementation are divided in: (i) water quantity and (ii) water quality. Although no traditional watershed-scale BMP modelling studies have been performed at the daily scale, a need to evaluate the effects of water quality sources and impacts is no less important in Barigui River basin, one of the most important for the cities of Curitiba and Araucaria and an affluent to a seven power-plant equipped river, the Iguassu River.

Since our goal is to achieve daily scale(data limitations) and in-situ representation, we chose SWAT running at the daily scale using the CN number as rainfall-runoff algorithm To overcome SWAT and CN method limitations, in addition to the others herein mentioned, simplifications are necessary to clearly state a hypothesis and boundary conditions to the problem:

(i) SWAT is assumed to correctly account geographically for water balances, in a sense that changing HRU-scale parameters will realistically account for flow reduction at the subbasin scale.

(ii) SWAT is assumed to encapsulate enough phenomena to describe watershed behavior, or at least, that the imperfections are negligible for water quality model purposes.

(iii) BMP performance does not change overtime.

(v) The CN number may be successfully used to evaluate BMP hydrological performance, considering both the model limitations, namely the assumption that overland flow equals total runoff, and literature CN values to represent field-scale urban BMPs.

This study will ultimately attempt to mimic pre-spatially allocated BMP flow and water quality conditions, neglecting landscape transmission and lumping its effects at the sub-watershed scale, using placement, sizing and performance simplifications, mainly based on GIS data processing or available literature.

The procedure to incorporate the land use effects is to directly insert modified land uses using the ArcSWAT input and re-parameterize these areas to reproduce BMP conditions.

Although many techniques and algorithms are available for BMP spatial allocation and optimization, this study will assume that BMPs may be place wherever they are feasible to be constructed, under the hypothesis that a comprehensive framework could enforce in situ stormwater management strategies and that feasibility criteria would remain similar. BMP allocation was performed using the SUSTAIN BMP Sitting Tool (Tetra Tech Inc.,2013). The tool is a part of the SUSTAIN package and runs as a plugin for ArcGIS 10.1. The tool manages and overlays a series of user-defined raster grids and features (DEM, LULC, HSG, Street layout, stream grid and land ownership), highlighting potential locations for urban BMPs. The BMP spatial feasibility criteria are synthesized in

Table 9, below:

Table 9 - BMP placement criteria

Wet pond	Drainage Area	Slope	Land Use/ Land Cover	HSG	Water Table depth	Road buffer	Stream buffer
Infiltration Basin	< 6 ac	<15%	All but agricultural	A-D**	>1.20m	NA	> 30m
	< 24,281 m²				> 1.22 m	NA	> 30 m
Grassed Swales	< 5 ac	<5%	All but agricultural	A-D	>0.61m	< 1.8 m	NA
	< 20,234 m²				> 0.61 m	< 1.83 m	NA
Bioretention	<2 ac	<5%	All but agricultural		> 0.61 m	< 30m	> 30 m
	> 101,171 m²			A-D	> 0.61 m	< 30.84 m	> 30 m

The representation of the location and area of BMPs is not sufficient to represent the BMP hydrological effects. The CN method original formulation computes total effective rainfall for an event, lumping all hydrological behavior using the CN and S parameters to correlate rainfall and runoff. With that stated, the definition of the BMP drainage areas is necessary to allow for a conceptually consistent representation using the Curve Number.

To allow the creation of drainage areas for BMPs, Swale features were manually restituted at the 1:3000 scale, to create a more realistic BMP scenario, as previous/impervious footprint maps are not available yet. For Infiltration Basin, every parcel that contained a device was considered to be existent. Biorretention devices were existent wherever they could exist, as long as the location did not require land expropriation. To effectively create the BMP drainage areas, the pySWATBMPApp application was developed. It is a GIS-coupled Python tool to resize the BMP polygon feature to the target influence area. The PySWATBMPApp is more thoroughly discussed in Section 4.8.4.

BMP spatial allocation provides evidence on what is the maximum BMP area ($ABMP_{max}$), as BMPs drain its own area plus a value that varies between its own area (i.e.: zero drainage area) and the totality of drainage area upwards to the BMP, according to the SWAT lumping method. Thus, determining a delivery area is important to adequately represent the BMPS. This work is assuming that every BMP is homogeneous sized for a certain rainfall volume, and hence an area of influence can be derived if an average depth is assumed. This method was selected due to its simplicity and capability of being implemented on GIS systems. This method was not copied from other work. In fact, to the best of the author knowledge, this method has not been yet implemented.

Mathematically, delivery area sizing will be calculated adopting the following assumptions:

(i) BMPs must hold a water quality control volume accounted as constant.

(ii) In-situ BMPs are not sized for large events or large detention volumes. Hence, the BMP volume is equal to hold the water quality rainfall volume relative to its influence area.

(iii) The BMP area is obtained from the feasibility GIS feature processing.

The calculation of the delivery area for Swales, as well as for the other BMPs is showed in Equation 27:

$$Darea_{Swale} = \quad Swale_{depth} * \frac{Swale_{area}}{Design_{rainfall}}$$ Equation 27

Where $\boldsymbol{Darea_{Swale}}$ is the delivery area to a swale, $\boldsymbol{Swale_{depth}}$ is the swale average depth. Since BMPs may be designed for various purposes(Kellagher et al., 2015), such as holding an initial runoff volume, for Swales, a **4 mm** retention volume will be adopted. Since swales are not deep, a **15 cm** depth is used for sizing. Swales are v-shaped conveyance channels. For this BMP, a maximum depth is computed as the average depth:

$$Darea_{Swale} = 0.15 \frac{Swale_{area}}{0.004}$$ Equation 28

As for infiltration basins, the same approach will be utilized. The delivery area will be calculated using a box-shape, as shown in Equation 30, adopting a maximum depth of 1.50m and a 2mm water quality volume.

$$Darea_{InfBasin} = min \begin{matrix} 1.50 \frac{InfBasin_{area}}{0.002} \\ AInfBasin_{max} \end{matrix}$$ Equation 30

Where $\boldsymbol{Darea_{InfBasin}}$ is the delivery area for an(m²), $\boldsymbol{InfBasin_{area}}$ is the Infiltration Basin feasible area (determined by the BMP sitting tool) and $\boldsymbol{AInfBasin_{max}}$ is the maximum drainable area by a single Infiltration Basin. This value is determined by setting maximum parcel drainage areas according with parcel area. These values were chosen after researching the dimensions of these devices. The selection of this criteria is based on sensitivity analysis and on the feasible practices for a realistic scenario at the city of Curitiba. Maximum parcel drainable area ratio is computer in four categories and showed in Table 10.

Table 10 - Infiltration Basin Maximum parcel area ratios

Maximum Drainable Area	Maximum Parcel Area
25%	<5,000 m²
20%	>5,000 m² and < 10,000 m²
15%	>10,000 m² and < 20,000 m²
10%	>20,000 m² and < 30,000 m²
3%	> 30,000 m²

The delivery area for Biorretention will be calculated using the same rationale. Assuming a maximum depth of 1.2m, and a 2mm design rainfall, Biorretention delivery area is calculated with Equation 29, below:

$$Darea_{Biorretention} = 1.20 * \frac{Biorretention_{area}}{0.002} \quad \text{Equation 28}$$

Where $\boldsymbol{Darea_{Biorretention}}$ is the delivery area for a Biorretention device (m^2), $\boldsymbol{Biorretention_{area}}$ is the raingarden feasible area (determined by the BMP sitting tool).

The reduced CN numbers, as proposed by Sample et al. (2001),using pre-defined CN values to assess BMP effects were originally used as inspiration for this work. However, single CN reduction values are not enough to evaluate watershed behavior during various infiltration-induced scenarios. To solve such a problem, this work assesses the watershed hydrological changes of using the BMPS in four CN reduction scenarios, in order to cover a range of CN reductions: **10%, 30%, 50%** and **70%**.

PySWAT, a series of Python programs were created for assisting the scenario reparameterization and input/output data visualization, topic described in the next section.

4.8.4 The PySWAT Program

Two algorithms were developed to assist in swale field-scale scenario generation - pySwatBMPApp, parameter changes, pySwatParApp and output analysis-pySwatGetI/O. The three-application bundle is PySWAT, a Python Framework for SWAT result analysis and BMP scenario generation.

The programs were developed in Python 2.7(pySwatBmpApp) and 3.6(pySwatParApp and pySwatGetOutput). The software is organized according to a framework structure (which means that the software is modular and scalable, and that a compiler and extra code are necessary for every usage). The application of the code to this case is limited to the needs of this work. The Annex section is structured to show the framework modules code and also the applications to this work. All codes are commented and available on GitHub. It is available on the PySWAT public repository(https://github.com/davidbispo/pySwat).

The PySwatBmpApp is based on the ArcGIS 10.3 Python interpreter – ArcPy. ArcPy provides automation and combination of various spatial analysis functions, starting from basic problems such executing buffer, erase and clip up to more sophisticated applications (ESRI, 2019). The program sizes BMPs according to pre-defined criteria, having three modules: (i) Swales, (ii) Infiltration Basins and (iii) Biorretention.

The Swales module performs two basic tasks: (a) mirroring the swales so they are located within the street polygon and (b) enlarging the features based on the street layout polygon until a desired drainage area is achieved. This is performed through transferring the BMP polygon to inside the street polygon, which is done through buffering the BMP feature, clipping the BMP feature using the street layout feature and adjusting the feature area using buffers. Required buffer for area reconstruction is calculated using the bisection method. Figure 15 shows examples of the mirroring and area reconstruction. In Annex 7 and 8 the source code for the swale mirroring can be found. The enlarging is performed through iterating buffer values and clipping the result features until a pre-defined value is achieved. The predefined value is given by Equations 28,29 and 30. The drainage area is the sum of the BMP area and the street area it drains, as depicted in the figure below.

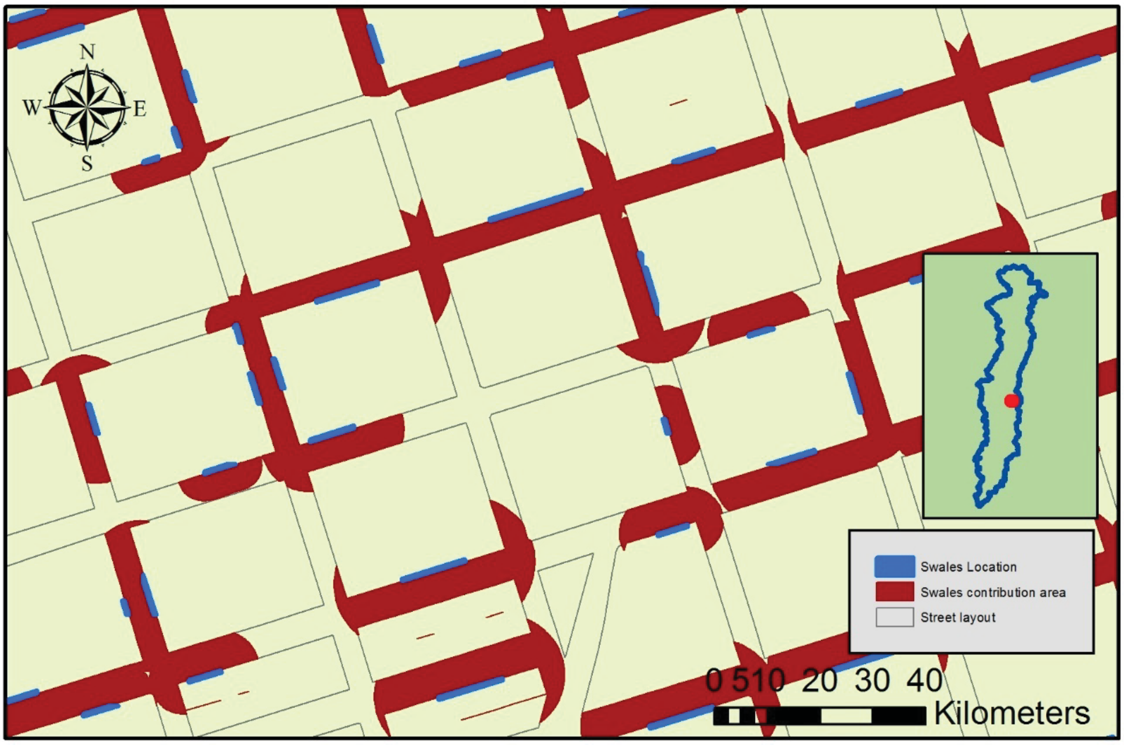


Figure 15 - Example of Swale placement using the PySwat BmpApp

The pySwatParApp is based on Python 3.6, using NumPy arrays for data extraction and parameter change. pySwatParApp uses as dependencies the dic_par library. It contains a dictionary (key-value pairs), where every string key (e.g.: CN2) corresponds to a SWAT input filetype (e.g. *.gw or *.hru) and the line where the parameter can be found. The pySwatParApp currently performs file-rewrite for all files that are within the (i) HRU structure (every HRU has a single file and the variable is in dic_par), (ii) SOIL structure and subbasin structures (e.g.: *.sub or *.rte files – files that define subbasin features. As long as key-value pair is in dic_par).

The pySwatGetIO is based on Python 3.6, using NumPy arrays for data extraction, Pandas for time-series analysis and matplotlib for data visualization. The pySwatGetOutput has three modules (i) SWAT model parameter reader - getpar, and (ii) SWAT model output reader – get_outtput.

get_par fetches all parameters for HRUs from a SWAT TXTInOut(Folder containing the input documentation for SWAT.exe run). It shows them as a numpy array, which suggests being used with either a JuPyter Anaconda terminal or an IDE such as Spyder. This application also uses as dependencies the **dic_par** library. dic_par is a contains Pythono module containing dictionary (key-value pairs), where every string key (e.g.: CN2) corresponds to a SWAT input filetype (e.g. *.gw or *.hru) and the line where the parameter can be found.

get_output is capable of reading *.rch files and *.hru files. The pySwatParApp currently performs file-rewrite for all files that are within the (i) HRU structure (every HRU has a single file and the variable is in dic_par), (ii) SOIL structure and subbasin structures (e.g.: *.sub or *.rte files – files that define subbasin features. As long as key-value pair is in dic_par).

5 RESULTS

5.1 Calibration and validation

The sensitivity analysis was performed as described in Section 4.8, and results for all the parameters as well as their description are shown in Figure 16.

#	Method	Parameter	Landuse	Subbasins	Min	Max	t-stat	p-value
1	r	CN2.mgt		40-98	-0.3	0.3	22.030	0.00
2	v	ALPHA_BF.gw	CMPO	40-98	0.05	0.3	-0.346	0.73
3	v	ALPHA_BF.gw	VARN	40-98	0.05	0.4	0.351	0.73
4	v	ALPHA_BF.gw	VARP	40-98	0.05	0.4	-0.338	0.74
5	v	ALPHA_BF.gw	AUBD	40-98	0.7	1	-1.154	0.25
6	v	GW_DELAY.gw		40-98	0.01	200	0.385	0.70
7	v	GWQMN.gw		40-98	0	5000	-3.953	0.00
8	v	RCHRG_DP.gw		40-98	0.2	1	4.017	0.00
9	v	DEEPST.gw		40-98	0	200	-0.258	0.80
10	v	SHALLST.gw		40-98	0	200	0.432	0.67
11	v	SLSOIL.hru		40-98	0	30	-0.558	0.58
12	v	LAT_TTIME.hru		40-98	3	15	-7.466	0.00
13	r	CH_N1.sub		40-98	-0.3	0.3	-1.166	0.24
14	v	CH_K1.sub		40-98	0	500	-9.006	0.00
15	r	CH_N2.rte		40-98	-0.3	0.3	-2.935	0.00
16	v	ALPHA_BNK.rte		40-98	0	1	23.386	0.00
17	v	CH_K2.rte		40-98	0	500	-36.927	0.00
18	r	SOL_AWC().sol		40-98	-0.3	0.8	-4.162	0.00
19	r	SOL_K().sol		40-98	-0.9	0.9	-1.675	0.09
20	r	CN2.mgt		99-155	-0.2	0.2	5.843	0.00
21	v	ALPHA_BF.gw	CMPO	99-155	0.05	0.3	-0.978	0.33
22	v	ALPHA_BF.gw	VARN	99-155	0.05	0.4	0.371	0.71
23	v	ALPHA_BF.gw	VARP	99-155	0.05	0.4	0.734	0.46
24	v	ALPHA_BF.gw	AUBD	99-155	0.7	1	-0.522	0.60
25	v	GW_DELAY.gw		99-155	0.01	200	-1.962	0.05
26	v	GWQMN.gw		99-155	0	5000	-1.688	0.09
27	v	RCHRG_DP.gw		99-155	0.2	1	-1.813	0.07
28	v	DEEPST.gw		99-155	500	5000	-0.807	0.42
29	v	SHALLST.gw		99-155	500	5000	0.310	0.76
30	v	SLSOIL.hru		99-155	0	30	-2.734	0.01
31	v	CH_K1.sub		99-155	0	500	-4.608	0.00
32	r	CH_N2.rte		99-155	-0.3	0.3	-4.792	0.00
33	v	ALPHA_BNK.rte		99-155	0	1	18.455	0.00
34	v	CH_K2.rte		99-155	0	500	-23.360	0.00
35	r	SOL_K(..).sol		99-155	-0.9	0.9	-0.704	0.48
36	v	SURLAG.bsn		All	500		-1.516	0.13

LEGEND

r - relative change[new = old * (1 + value)]	p-value<0.05 - Selected
v - replace change	p-value>0.05 - Void - Selected
v - replace change	p-vaue>0.05 - Not important after iteration

Figure 16 - Sensitivity analysis results

Sensitivity analysis shows that data is adequate for modelling, as climate, soil and flow data may not be always feasible for modelling in SWAT. Initial NSE values for standard parameter simulations were $NS_{precal-SQ} = 0.40$ and $NS_{precal-CX} = 0.45$.

In SQ, groundwater parameters, lateral travel time and reach routing/storage parameters showed significance, which was expected due to the urban nature of land uses. SLSOIL (Length groundwater flow inclined reservoir) did not reach enough p-value significance as groundwater flow is significantly reduced because of soil compaction and impermeabilization. Figure 17 shows observed and simulated values after calibration.

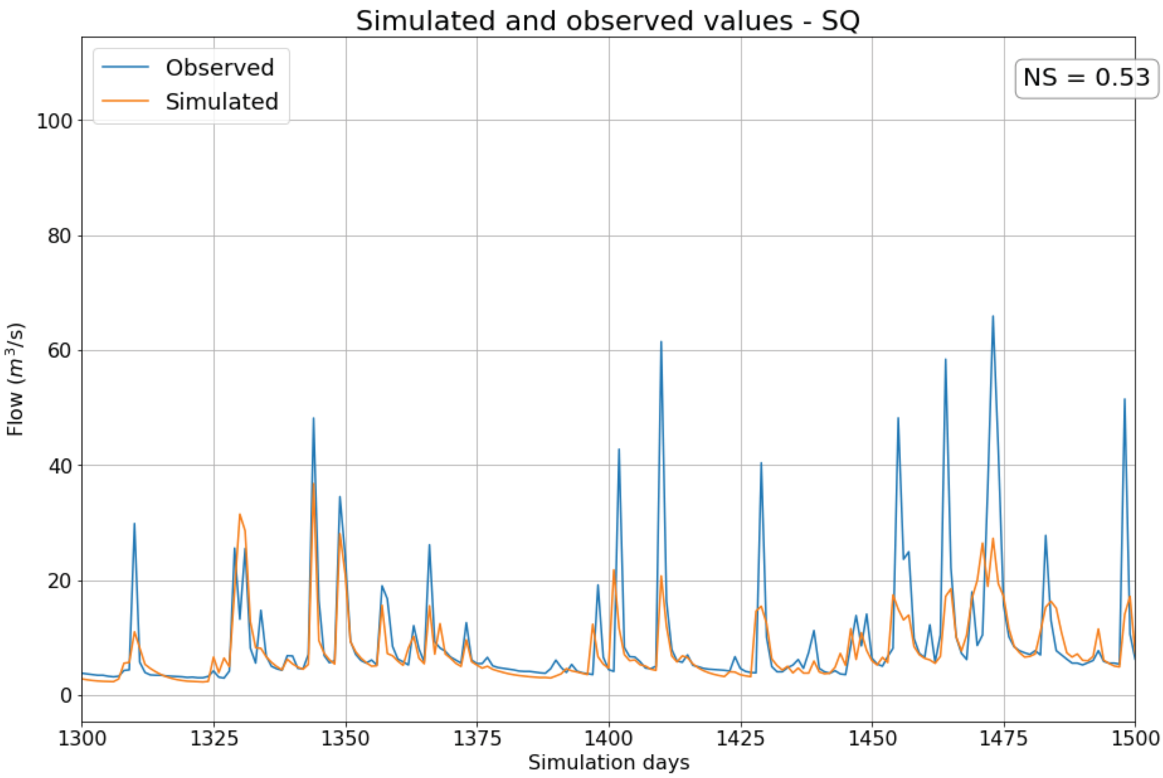


Figure 17 - SQ station simulated and observed values

In CX station, routing and groundwater parameter showed larger significance, including the GW_DELAY (HRU constant for computation of next time step flow delay effects). CN2 (initial CN value) for the SQ and CX parameters was not significant for model efficiency, which evidences good model responses for the basin and good parameter discretization and initial estimation. Figure 18 shows observed and simulated values after calibration.

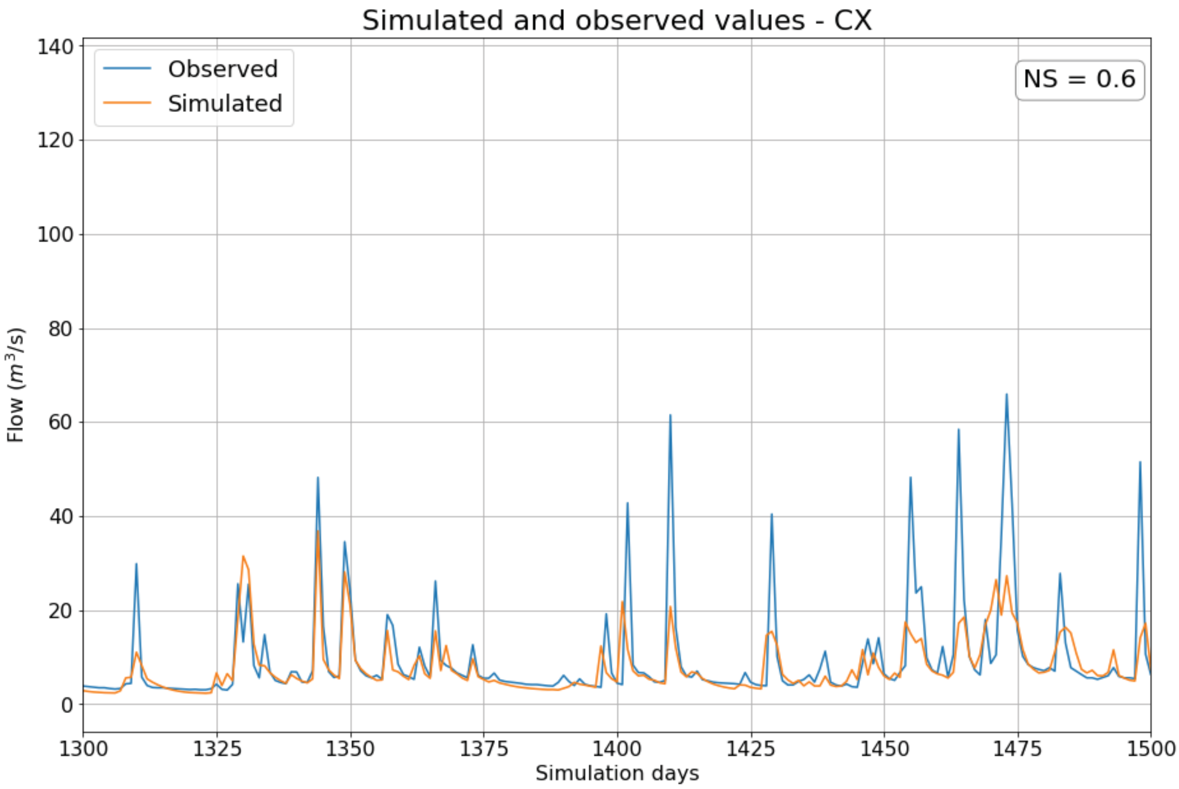


Figure 18 - CX station simulated and observed values

The ALPHA_BF different parameterization for urban and nonurban land uses did not show significance for any of the incremental areas, nor studied separately as a single coefficient to be applied to the totality of incremental areas.

The initial and final ranges for calibrated values, as well as its statistics, are shown in Figure 19, below.

Method	Parameter	Subbasins	Initial range		Final range		t-stat	p-value	Optimal
			min	max	min	max			
v	GW_DELAY.gw	40-98	0	500	-29.32(zero)	66.415	-0.105	0.916	20.410
v	GWQMN.gw	40-98	0	5000	5271.791	8753.604	0.189	0.850	6652.320
v	RCHRG_DP.gw	40-98	0.6	1	0.794	0.932	0.788	0.432	0.852
v	SLSOIL.hru	40-98	0	90	7.196	15.236	-1.077	0.283	10.120
v	LAT_TTIME.hru	40-98	10	20	-0.978209(zero)	1.838	-1.682	0.094	0.120
v	CH_K1.sub	40-98	0	500	-114.82(zero)	92.282	-5.863	0.000	18.663
r	CH_N2.rte	40-98	-0.2	0.2	0.219	0.386	-1.492	0.137	0.266
v	ALPHA_BNK.rte	40-98	0.1	1	0.369	0.790	3.553	0.000	0.556
v	CH_K2.rte	40-98	0	500	40.684	122.469	-1.343	0.181	68.660
r	SOL_AWC().sol	40-98	-0.3	0.3	0.176	0.614	-2.255	0.025	0.494
r	SOL_K().sol	40-98	0	500	-0.017	0.280	1.085	0.279	0.113
v	GW_DELAY.gw	99-155	0	500	0.000	20.008	-1.114	0.267	11.556
v	GWQMN.gw	99-155	0	5000	411.701	2037.949	0.487	0.627	1098.511
v	RCHRG_DP.gw	99-155	0.1	1	0.021	0.346	-1.467	0.144	0.187
v	SLSOIL.hru	99-155	0	90	0.000	15.178	0.214	0.831	6.548
r	CH_N2.rte	99-155	-0.2	0.2	0.045	0.044	-0.246	0.806	0.046
v	ALPHA_BNK.rte	99-155	0	1	0.788	1.097	1.453	0.148	1.242
v	CH_K2.rte	99-155	0	500	140.632	297.472	-0.068	0.946	202.652
r	SOL_K().sol	99-155	-0.3	0.3	0.340	1.142	-1.126	0.262	0.995

LEGEND		
r - relative change[new = old * (1 + value)]	(yellow)	p-value<0.05
v - replace change	(blue)	void values
v - replace change	(orange)	0.1 <= p-value <= 0.2

Figure 19 - Final parameter values and ranges

After four 400 simulations (the 500 recommended by the manual has proved to be excessive, showing zero values of p-values), the final parameters had high significance values, which states that ranges are significantly varying concerning the multiple answers of the models. Final calibrated efficiency values were for the SQ station NSE = 0.53 and R^2 = 0.54. The CX flow station presented efficiency values of NSE = 0.60 and R^2 = 0.61.

5.2 Final BMP Layout

Based on the outputs of the BMP sitting tool and contribution area criteria proposed in Section 4.8.4, BMP drainage areas were sized and constructed as Land Use Input Features using the pySwatBmpApp. Processing times were extensive, taking up to 1 week of processing for Infiltration Basins under favorable processing speed and memory conditions.

It should be pointed that the scenarios generated in this study are purposefully exaggerated. The economic feasibility of such several devices, is considered negligible. This works aims to assess and quantify the hydrological effects caused by the implementation of BMPs. Preliminary results showed it could be necessary to push the results to assess its changes in terms of quantity enhancements. The construction of Swale features resulted in a total of

5612 Features, covering a total area of 5,958 km^2, which represents 2.54% of the watershed area. Swale densities varies from 0 Swales to 237 Swales in Basin 95 area. Figure 20, shows the final BMP layout for Swales. Figure 21 shows the heatmap density for Swales.

Infiltration Basin feature processing resulted in 79415 features, covering a total area of 37.80 km^2, which represents 14.34 % of the basin area. The Barigui River basin in the city of Curitiba currently has 185,222 parcels (42.87% of the watersheds parcels are covered by Infiltration Basins). Figure 22 shows the Infiltration Basins Layout for the modified CN Simulations.

Figure 23 shows a density heatmap for Infiltration Basins. Its densities varied from 0 Infiltration Basins/km^2 in subbasin 66 region, close to Almirante Tamandare Border, to 3504 Infiltration Basins/km^2 in subbasin 95 region. Infiltration basins, present within parcels as defined earlier, are present in 36.3% of the watershed parcels. As for Biorretention, a total of 3552 features were generated, covering a total area of 16.55 km^2, which represents 6.27 % of the basin area. Figure 24 shows the final layout of Swales in the Barigui basin, also depicting a few zoomed regions for better visualization. Figure 25 shows a density heatmap for Biorretention devices. Densities vary from 0 Biorretention basins/km^2 to 175 Biorretention basins/km^2.

It should be noticed that the number of BMPs placed is very high compared to what would be realistically feasible to implement on the watershed. Also, great computational effort was required to place the BMPs according the exposed method. The goal is to verify an extreme scenario, and if necessary, pinpoint the effects of BMP implementation starting with an extreme scenario.

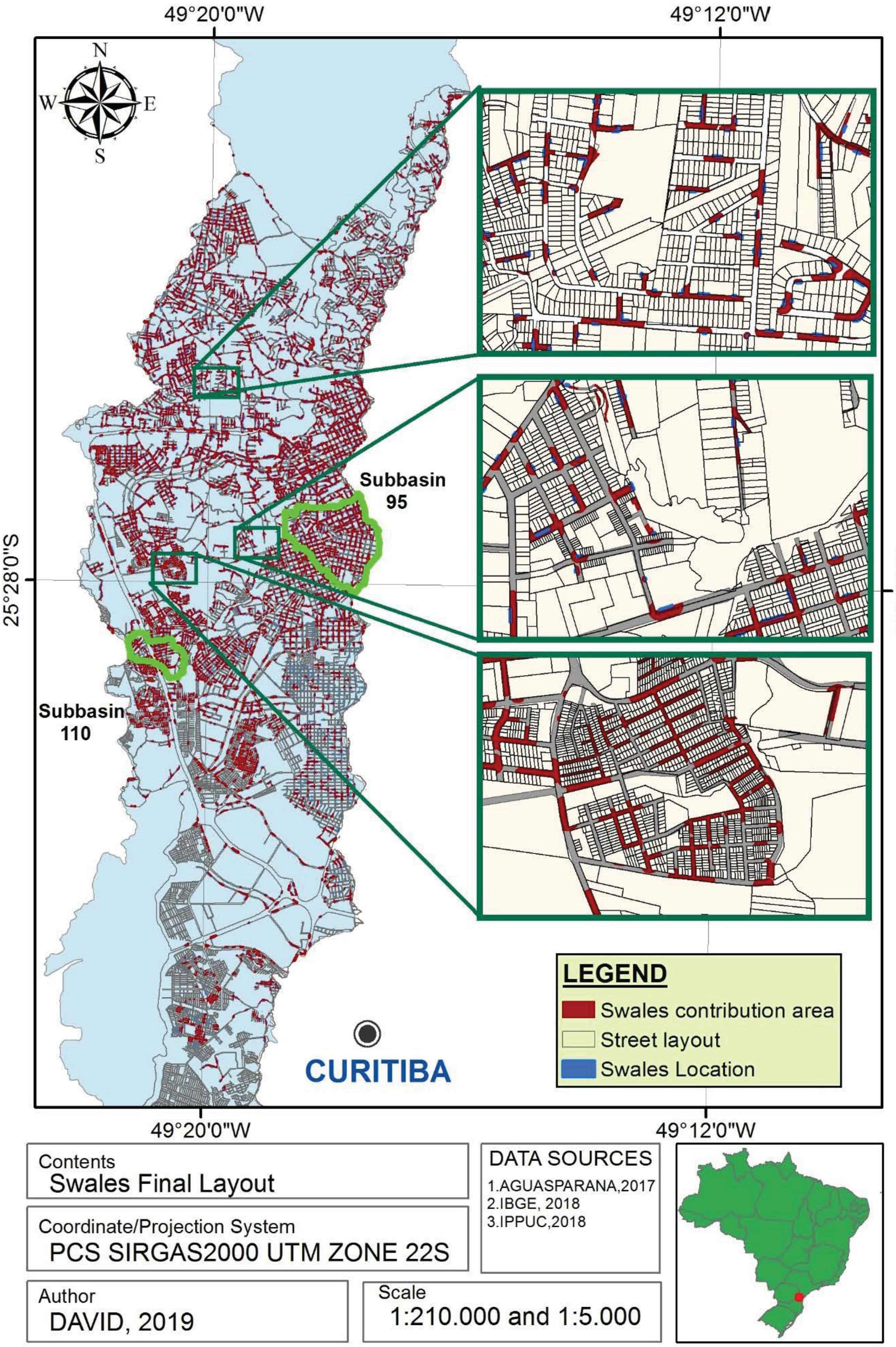


Figure 20- Swales Layout

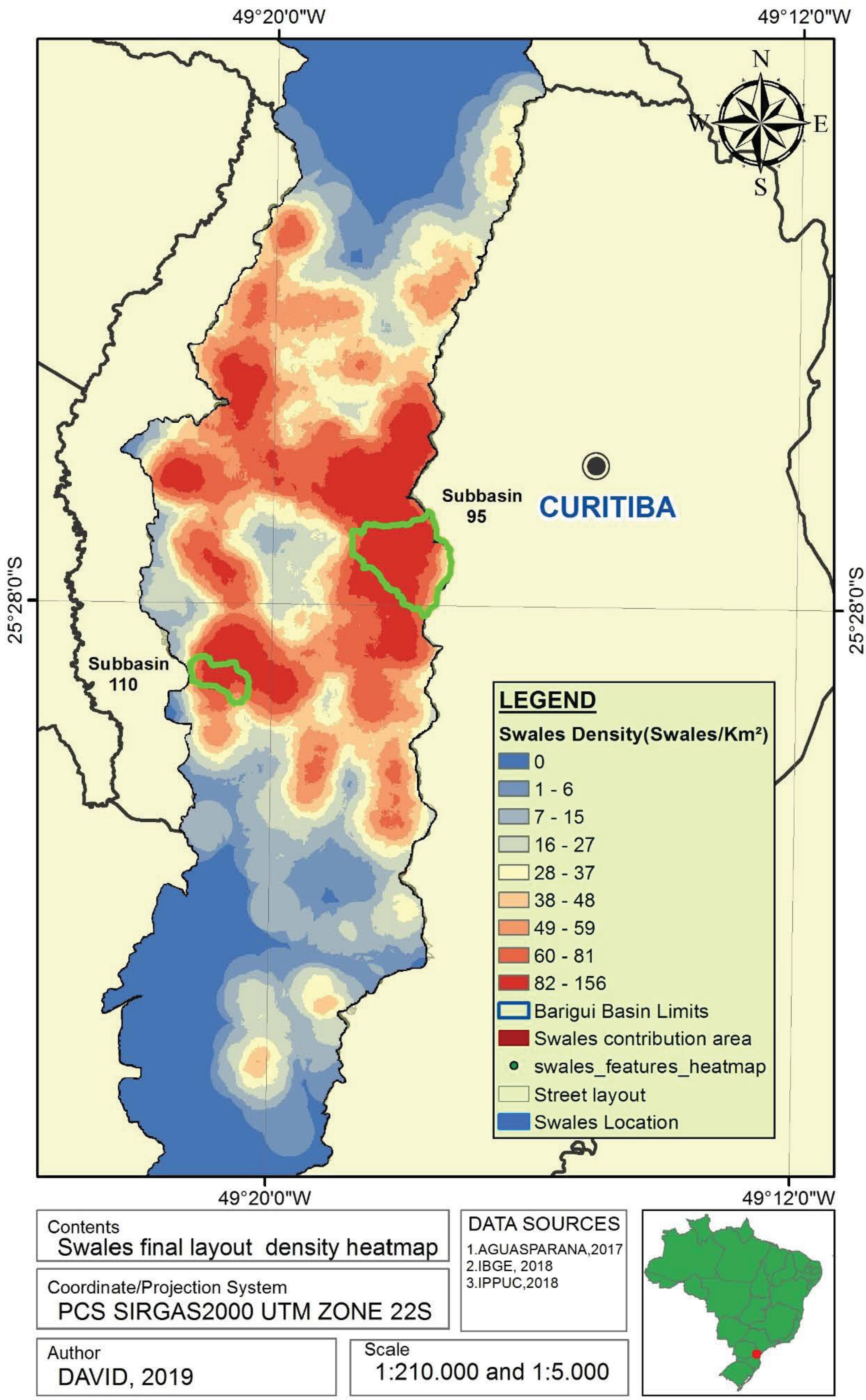


Figure 21 - Swales Area Heatmap

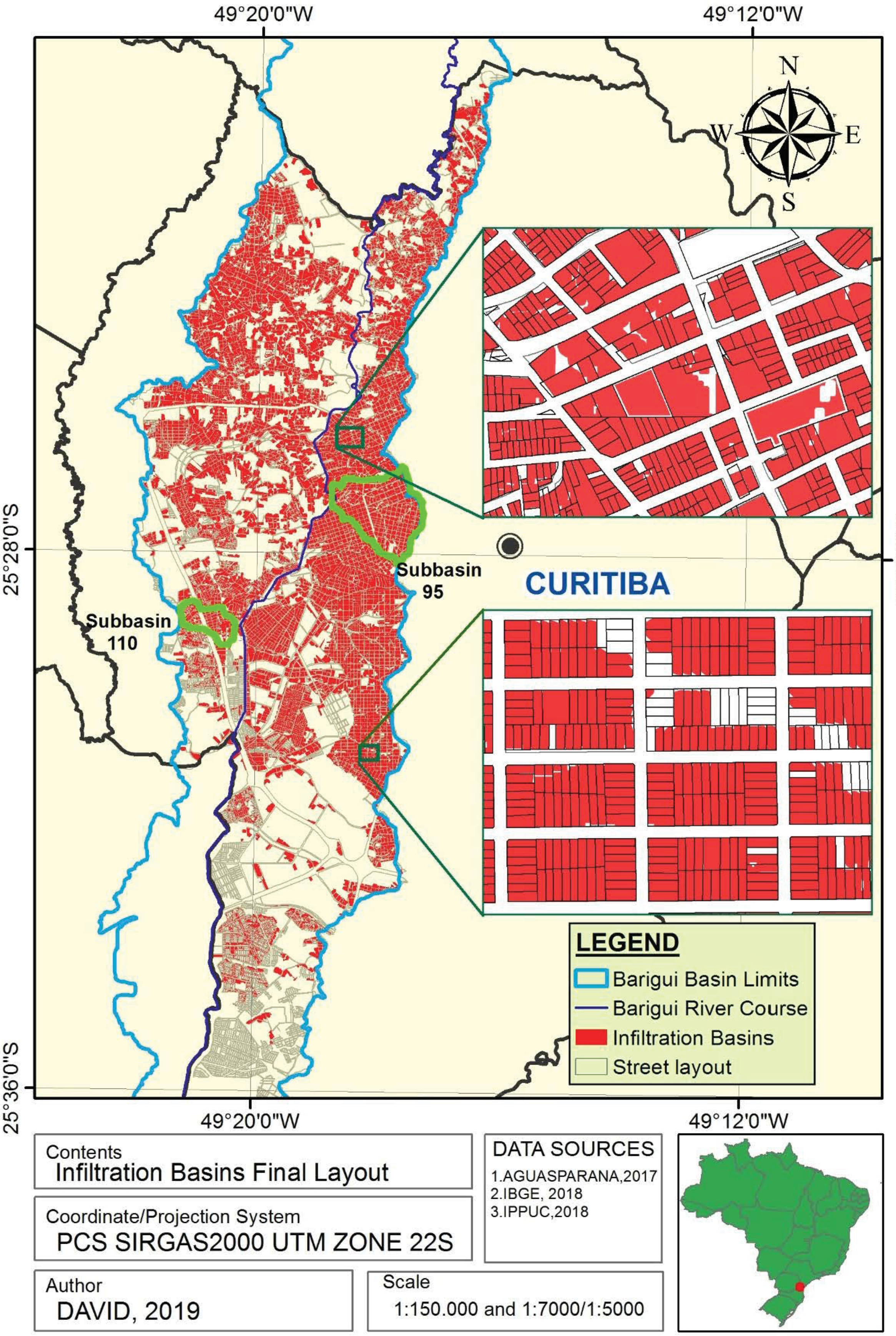


Figure 22 - Infiltration basin layout

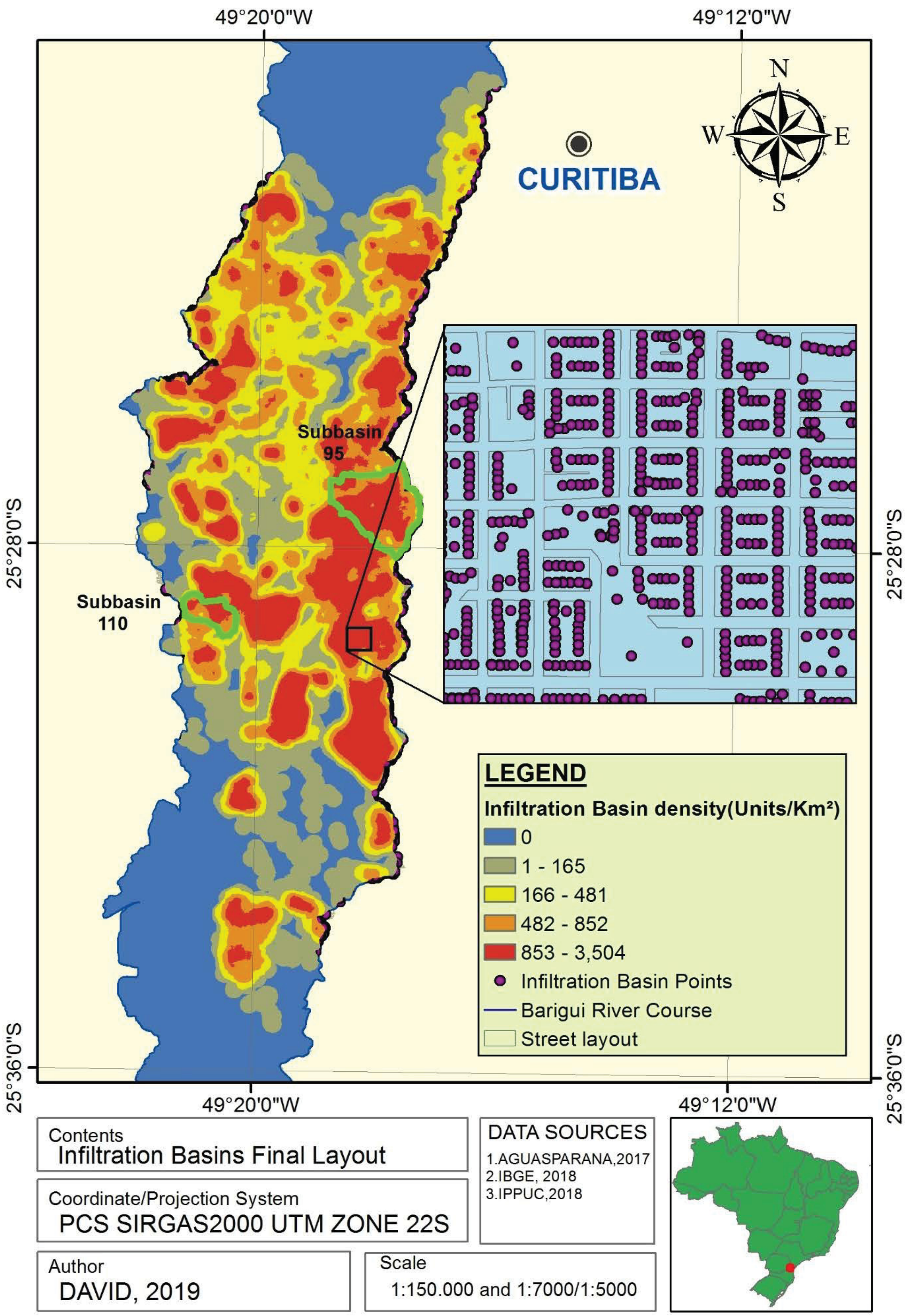


Figure 23 - Infiltration basin density heatmap

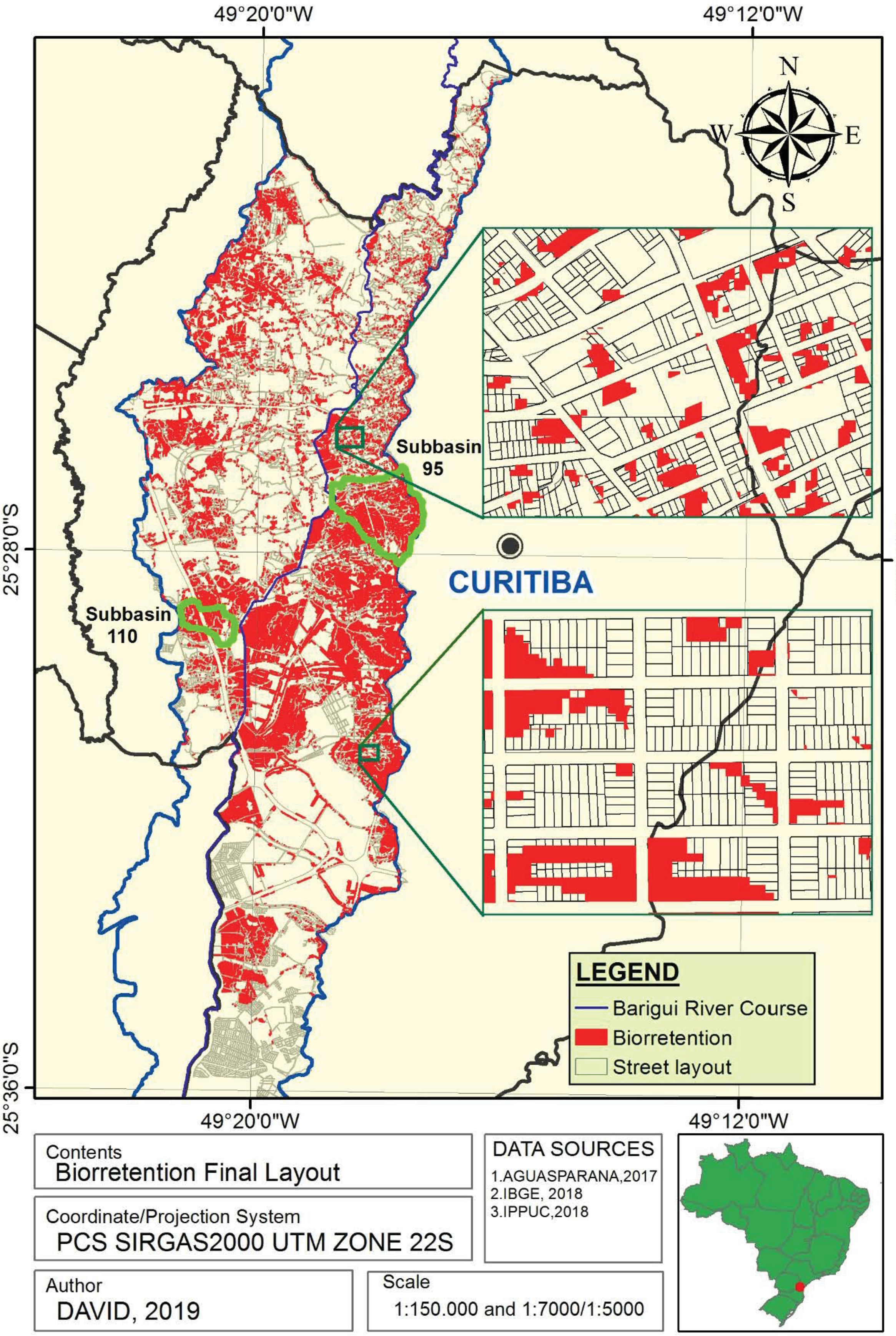


Figure 24 - Biorretention layout

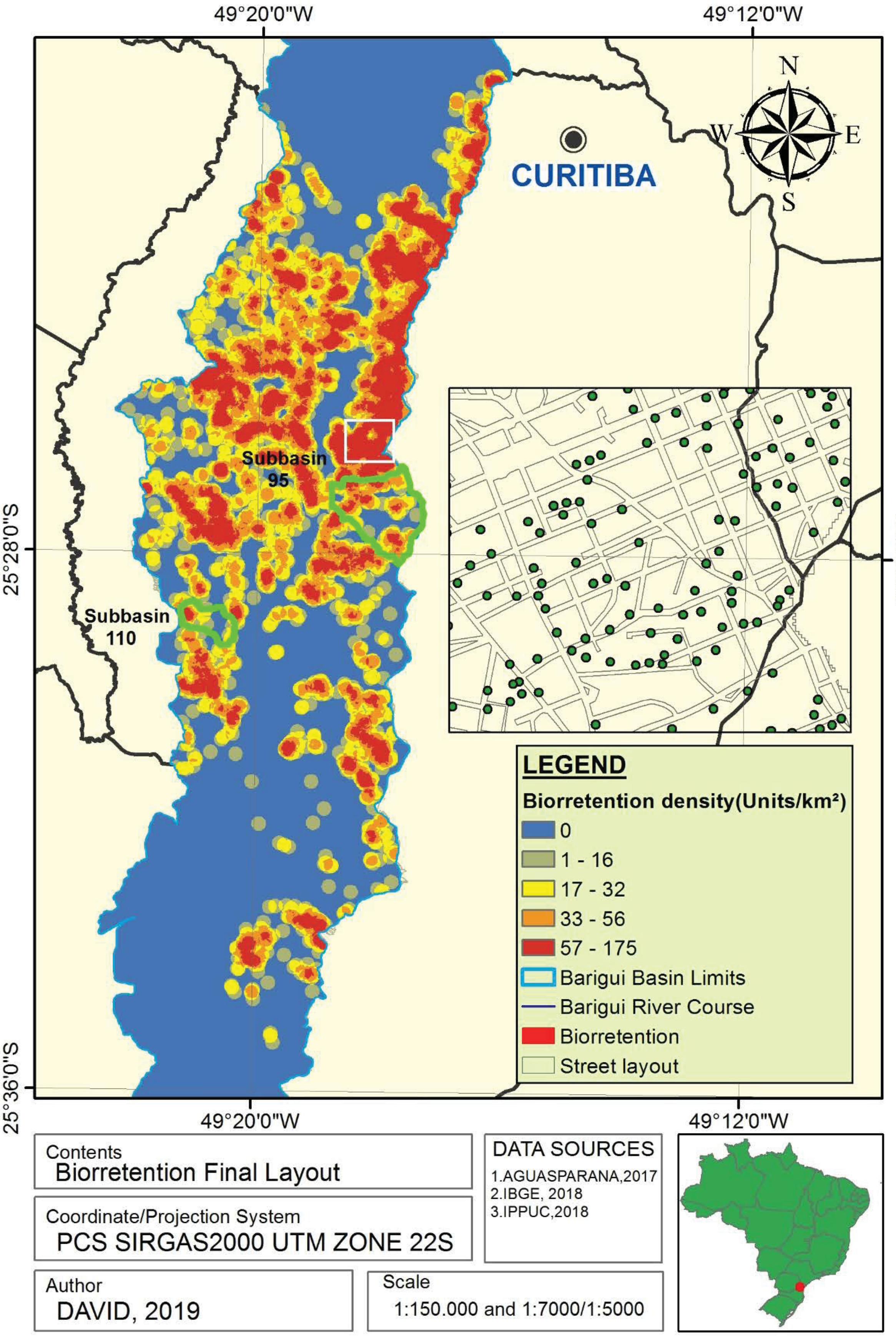


Figure 25 - Biorretention density heatmap

5.3 BMP hydrological effects

SWAT outputs various hydrological simulation results for all its computation scales, notably HRU, Subbasin and Reach. At each level, water balances are calculated for HRUs and grouped into subbasins. At the subbasin level, daily mass balances are performed and at each interval reach flow is calculated, from the Water Yield (WYLD) parameter, which is the sum of Lateral Flow (LAT_Q), Surface Runoff (SURQ) and Groundwater Flow (GW_Q), as stated in line 411 of the "**subbasin.f**" of the SWAT revision 664 source code (NEITSCH et al.,2011).

In this study, the 2401 HRUs behavior for all variables involved during all fourteen years of simulation will not be examined, as it demands special techniques for big data analysis. As an example, the "output.hru" files are 1.9 Gb and cannot be handled by common text, spreadsheet or even *in-array* structures, such as Numpy or Pandas, Scientific Analysis packages also built in Python. As a shortcut, a two-point strategy was constructed: (i) Using SWAT Output Viewer (YU,2015) to compile model results - The program is composed of SWAT-SQLITE, an application that runs the SWAT binary file directly into an SQLite Database and compiles results for all output scales, allowing resampled time-series analysis (i.e.: analysis of daily, monthly and annual values at all SWAT scales) and (ii) Not analyzing HRU-scale time series and its individual effects. Due to time limitations, a simple sampled HRU-Subbasin and Annual Watershed coupled analysis will be performed to assess the HRU and subbasin sensitivity to BMP implementation and watershed general effects.

5.3.1 BMP effects: SWALES

Swales effects on hydrological conditions are mainly dependent on underlying soil layer depths, GIS morphological HRU inputs (i.e.: soil slope) and HRU parameterization, which increase mass transfers from SWAT reservoirs to WYLD. To depict typical BMP behaviors, two HRU hydrographs in different subbasins, equipped with BMP devices were selected: HRUs 1342 and 1540.

HRU 1342 is in Subbasin 95 with underlying soil of the URBAN type, which is a one-layer soil profile. The HRU occupies a total area of **0.508 km^2 (12.96%)** of subbasin area, and its original land use is residential. The area was chosen due to its large BMP feasibility area and it depicts a common combination of soil and land use, besides a significant portion of land coverage within its subbasin. Figure 26 and Figure 27 show multi-variable plots for the 10% and 30% reduction scenarios.

During the simulation, HRU 1342 shows lower SURQ peak values and higher Shallow Aquifer Storages (SA_ST) and Deep Aquifer Storage (DA_ST) values were observed. In general, water balances showed no reduction for Soil Water (SW) and Groundwater Flow (GW_Q). As for variables that presented variation, Surface Runoff (SURQ) and Water Yield (WYLD) showed decreases on peaks. The Lateral Flow (LAT_Q) showed a slight increase during peaks.

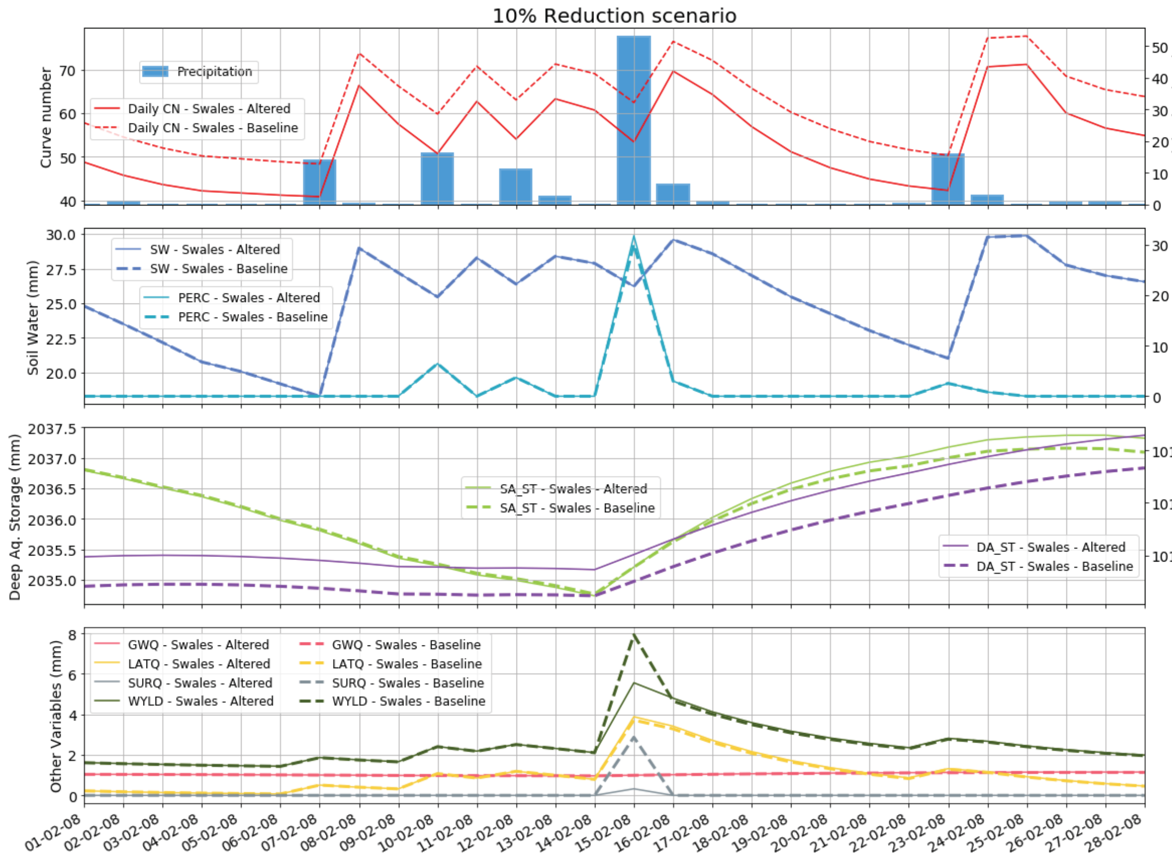


Figure 26 - 10% CN Reduction Scenario Hydrograph - HRU 1342

Differences can be noticed between the baseline scenarios, 10% reduction and 30% reduction scenarios. No difference was observed from 50% and 70% scenarios to the 30% scenario. Its graphs can be found in Annex 14 and Annex 15. The model calibration suggests for the TM-SQ incremental basins low Lateral Travel Times (LAT_TTIME, ranges from 0-20) and high Soil Available Water Content (SOL_AWC) than originally estimated. As these two variables are calibrated to be relevant for the HRU WYLD, according to the SWAT formulation, Lateral Flow (LAT_Q) is bound to be one of the driving forces for Water Yield (WYLD) values. Also, since URBAN type soils are parameterized with a single 30 mm soil layer, only one soil layer is working as a SWAT soil reservoir. Hence, soil moisture levels are frequently above Field Capacity, and LAT_Q, DAILYCN, and WYLD become less sensitive to soil moisture, responding very sharply to it, transferring water from the soil profile either as Lateral Flow or Aquifer (both shallow and deep). During larger events, Daily Surface Runoff (SURQ) is significantly reduced and LATQ increased-reaching constant value due to soil saturation. Despite changes in Soil Water (SW), percolated water increases are very small (~1mm) both for the Shallow Aquifer and Deep Aquifer.

Concerning Shallow aquifer storage, the effect produced are faster and larger storage values. As for the Deep Aquifer, effects long term-wise showed higher storage levels than the baseline scenario. As soil moisture controls directly the HRU Water Yield (WYLD) and the water table does not change significantly in storage, GW_Q remains constant during the simulation. Both WYLD and SURQ values do not change for the 50% and 70% scenarios as infiltration reaches a constant value, and WYLD does not increase since water cannot enter the soil profile after full saturation. Finally, although changes can be seen in HRU hydrograph in SURQ, and WYLD and LAT_Q, the surface runoff removed is mainly distributed between lateral flow and the deep aquifer.

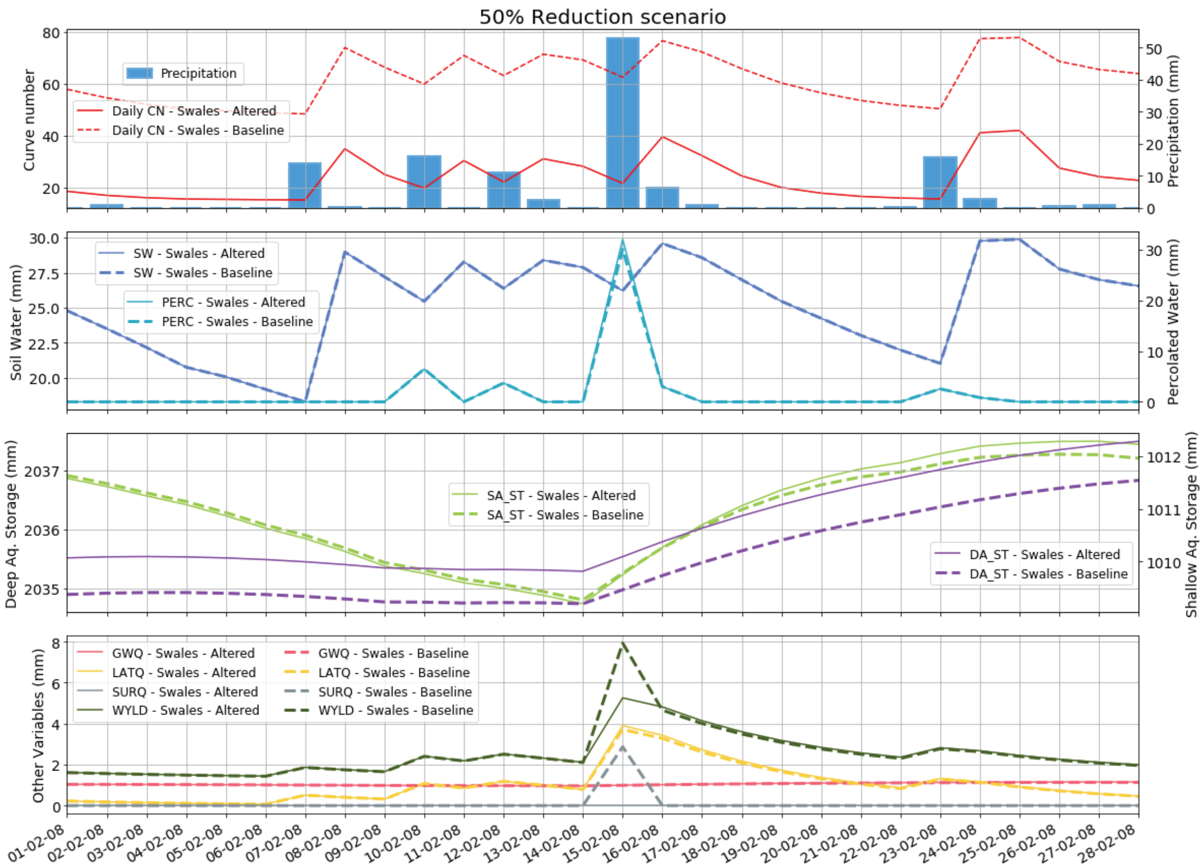


Figure 27 - 50% CN Reduction Scenario Hydrograph - HRU 1342

Changes can be seen in the HRU water balance, as expected for typical water balances. On the Subbasin Scale, however, water balances do not change sufficiently to cause noticeable effects in the water yield on the subbasin, and hence, to the reach flow, as shown in Figure 28 and Figure 29. Except in very extreme events, where balances are significantly affected by surface runoff, no significant changes can be noticed for none of the four scenarios.

For Subbasin 95, the average annual soil water increased from 29.1 mm to 32.1 mm (9%), with an annual surface runoff changing from 464 mm to 235mm (97.4%). Groundwater flow annually varied from 267 mm to 323 mm. Lateral Flow decreased from 335 mm to 317 mm. Despite the significant changes on groundwater and surface flow, water yield values did not vary. As one would expect, channel flow also did not change, despite all the compartment changes taking place in the subbasin 95 with the addition of BMPs, or the reduced CN areas hypothesized to work as BMPs.

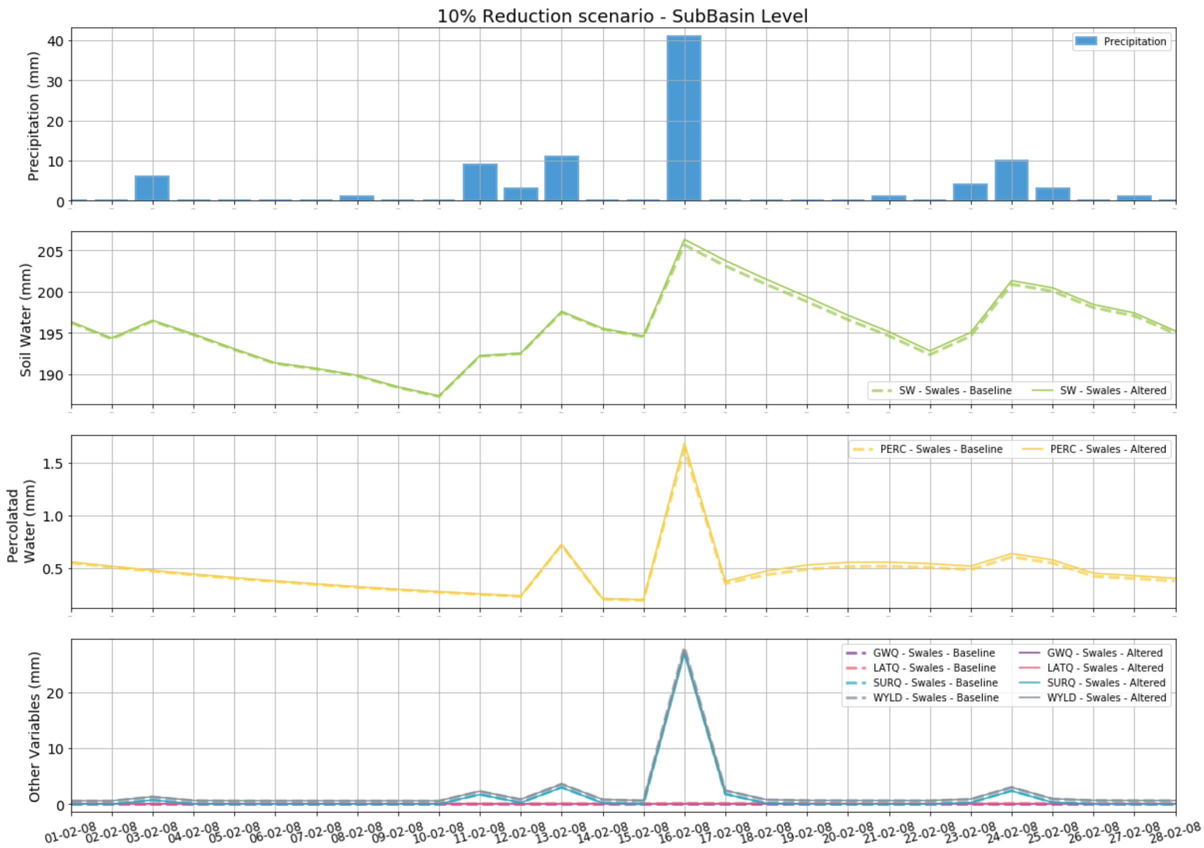


Figure 28 - 10% CN Reduction Scenario Hydrograph - Subbasin 95

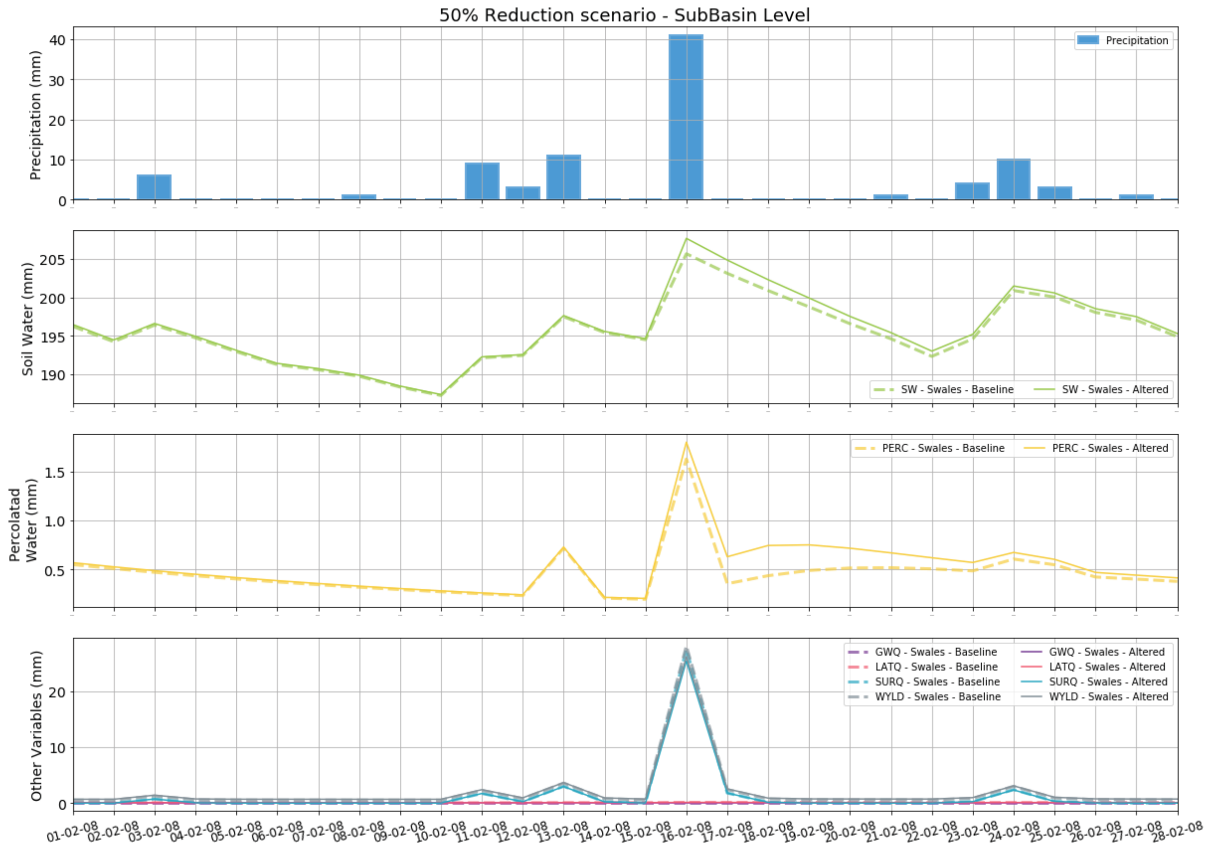


Figure 29 - 50% CN Reduction Scenario Hydrograph - Subbasin 95.

Another typical Residential Land Use HRU was chosen to illustrate the effects of BMPs in non-typically soils, HRU 1554, located in Subbasin 110. The HRU occupies **0.098 Km^2 (61% of Subbasin area)** and is located in an ARGISOIL, which has 6 layers, typically with values of SOL_K lower than 100 mm/h. The six layers ought to grant more storage and transmission potential between soil layers and from soil to the Shallow and Deep Aquifers. The calibrated value for RCHRG_DP suggests that precipitated water percolates and it is transformed in Deep Aquifer Storage (it is lost from the system) rather than Lateral Flow (LAT_Q) or Groundwater contribution to WYLD (WYLD).

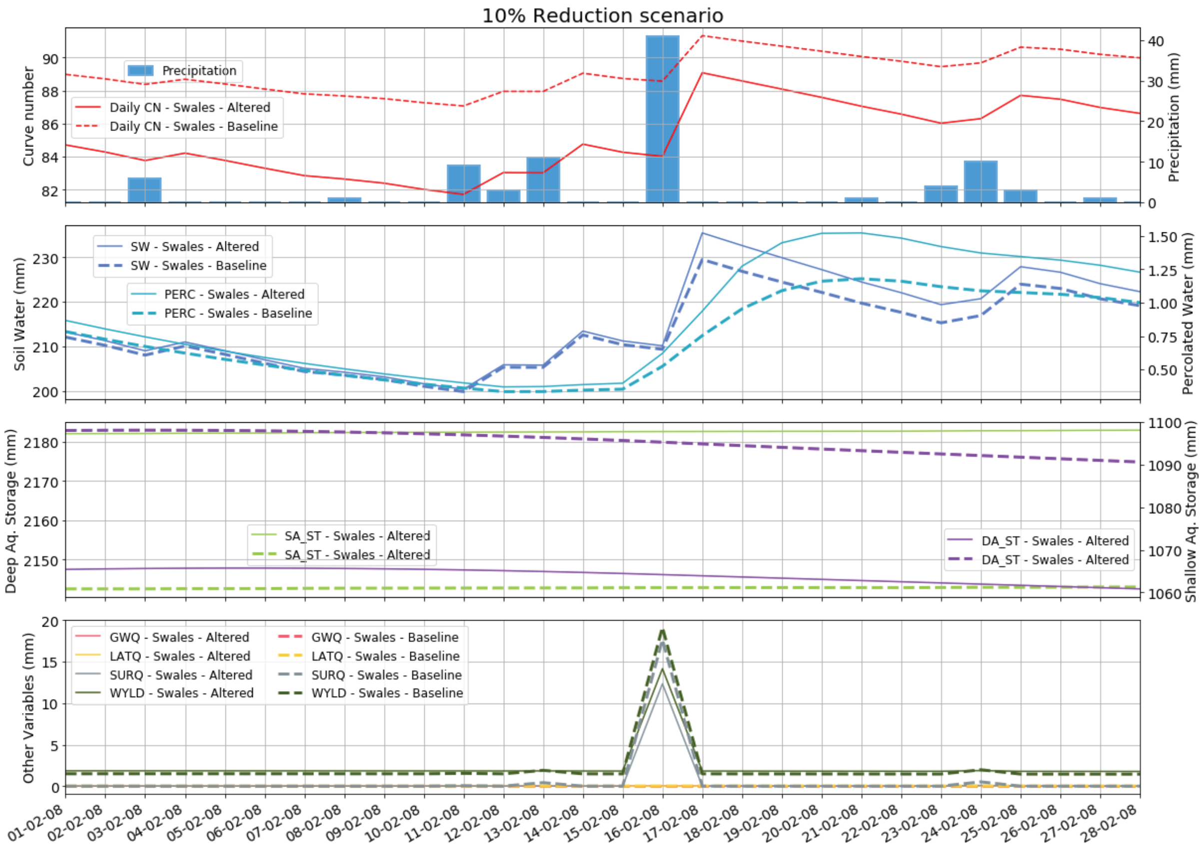


Figure 30 - 10% CN Reduction Scenario Hydrograph - HRU 1554

Figure 30 and Figure 31 depict the HRU-level reductions obtained for the 10% and 50% reduction scenarios in HRU 1554. In both cases a more hysteretic system reaction can be observed, as SW values increase and CN values tend to be shift more abruptly during peak events only, contrary to the observed on the Baseline scenario. PERC values increase significantly during rainfall events. Consequently, DA_ST and SA_ST increase, especially DA_ST, as the subbasins downstream to 99(SQ-CX) are calibrated with very high Shallow and Deep Aquifer partitioning values (RCHRG_DP=0.943) value, which means that 94.30% of the percolated content is transferred to the Deep Aquifer (the remaining is routed to the Shallow Aquifer). Even though, it can be noticed that the increase in the Shallow Aquifer is significant, and some water is being lost in the deep aquifer during the studied event.

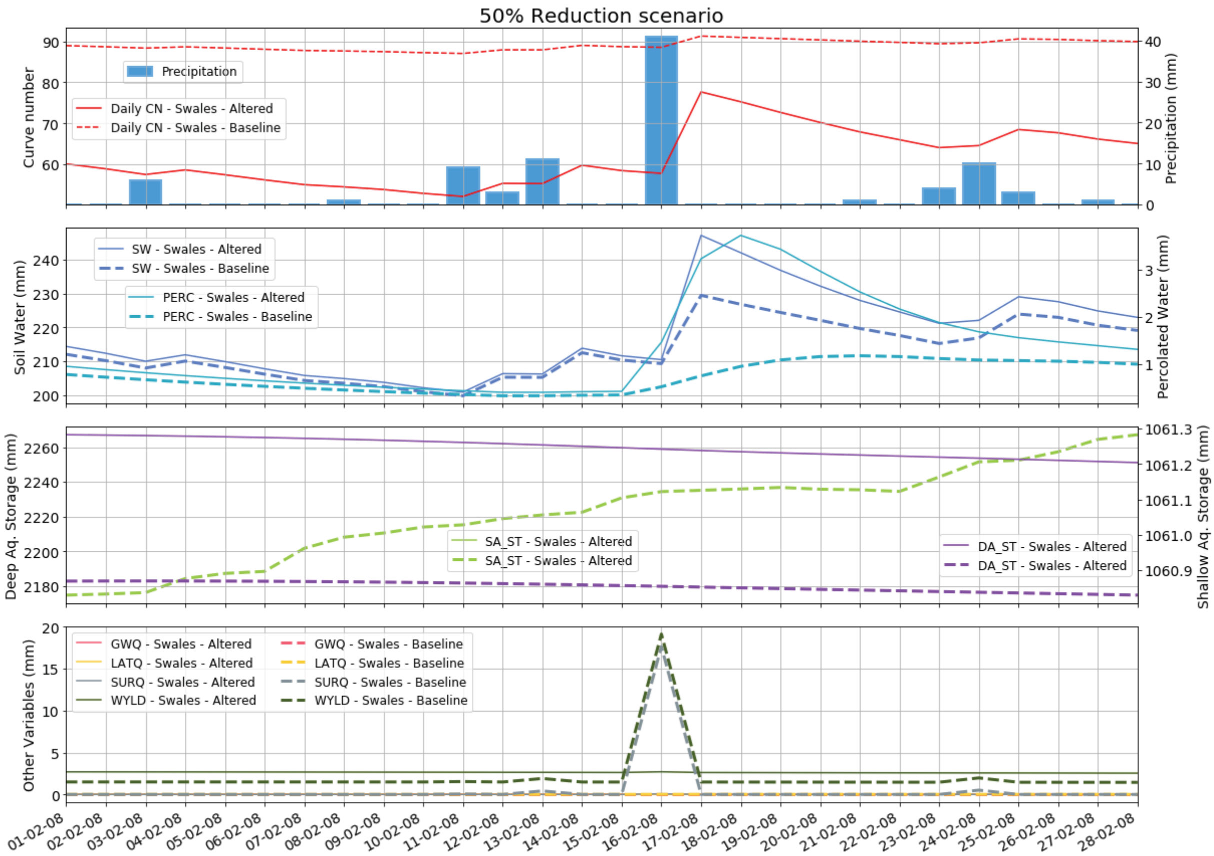


Figure 31 - 50% CN Reduction Scenario Hydrograph - HRU 1554

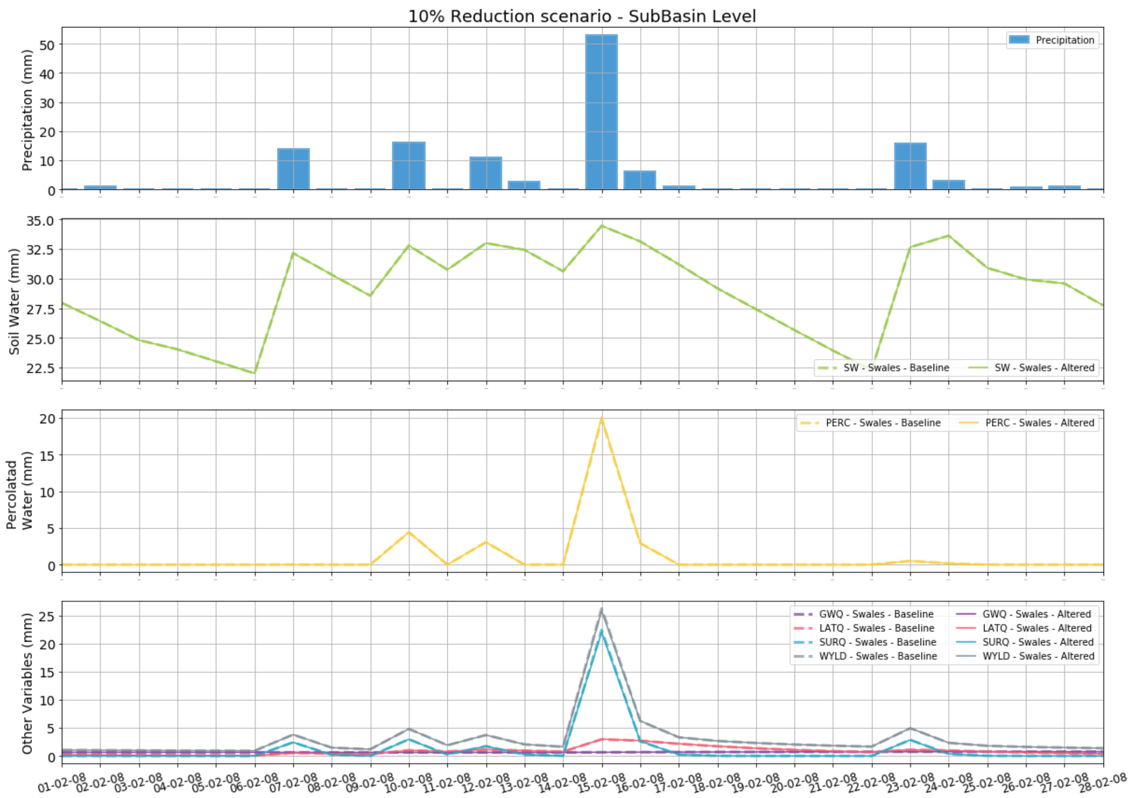


Figure 32 - 10% CN Reduction Scenario Hydrograph – Subbasin 110

Although visible changes are noticeable at the HRU scale, no significant changes can be noticed at the subbasin scale. The entire subbasin response to the precipitation event is not changed by the creation of lower CN zones. In other words, WYLD component do not contribute sufficiently to change its values during the event.

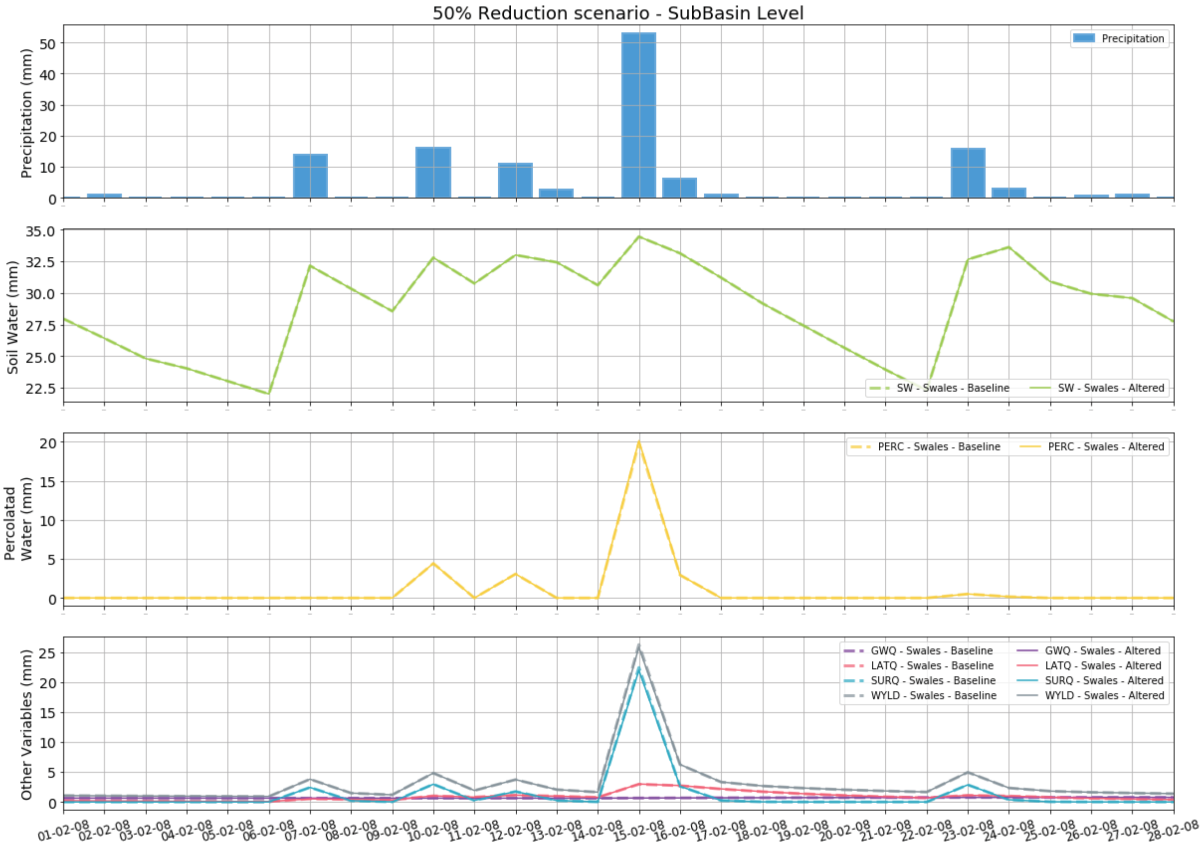


Figure 33 - 50% CN Reduction Scenario Hydrograph – Subbasin 110

Despite the changes on CN for all four scenarios, results showed that no significant changes in any parameters are noticeable during the simulation for Subbasin 110. No significant changes are observed in SW, PERC, GWQ, LAT, SURQ or WYLD, with values remaining constant during the simulation, when compared with the Baseline Values. Figure 34 depicts in the subbasin scale the annual mean reductions during the Simulation Years (200-2014), not counting the warm-up period.

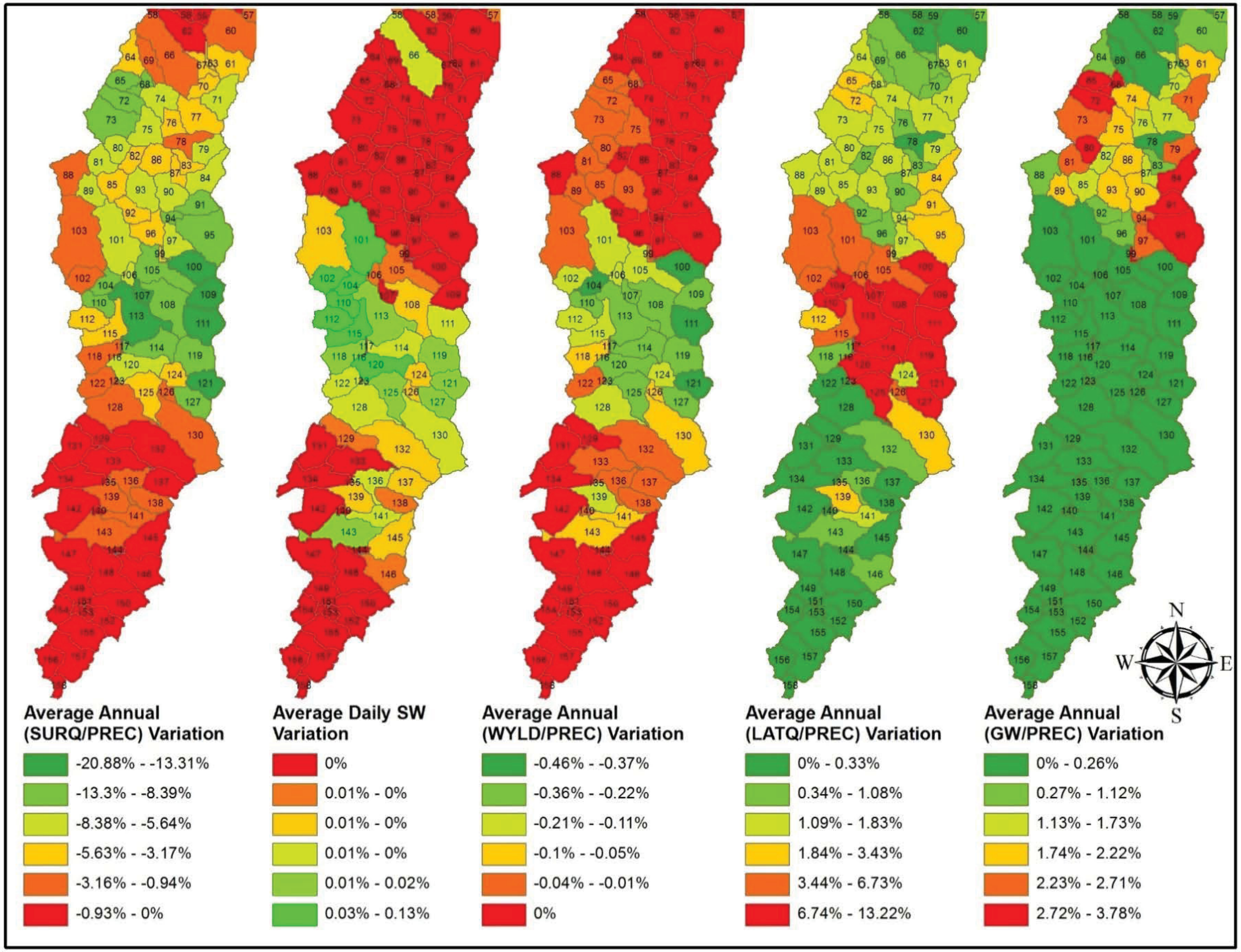


Figure 34 - Swales 10% Watershed Reductions

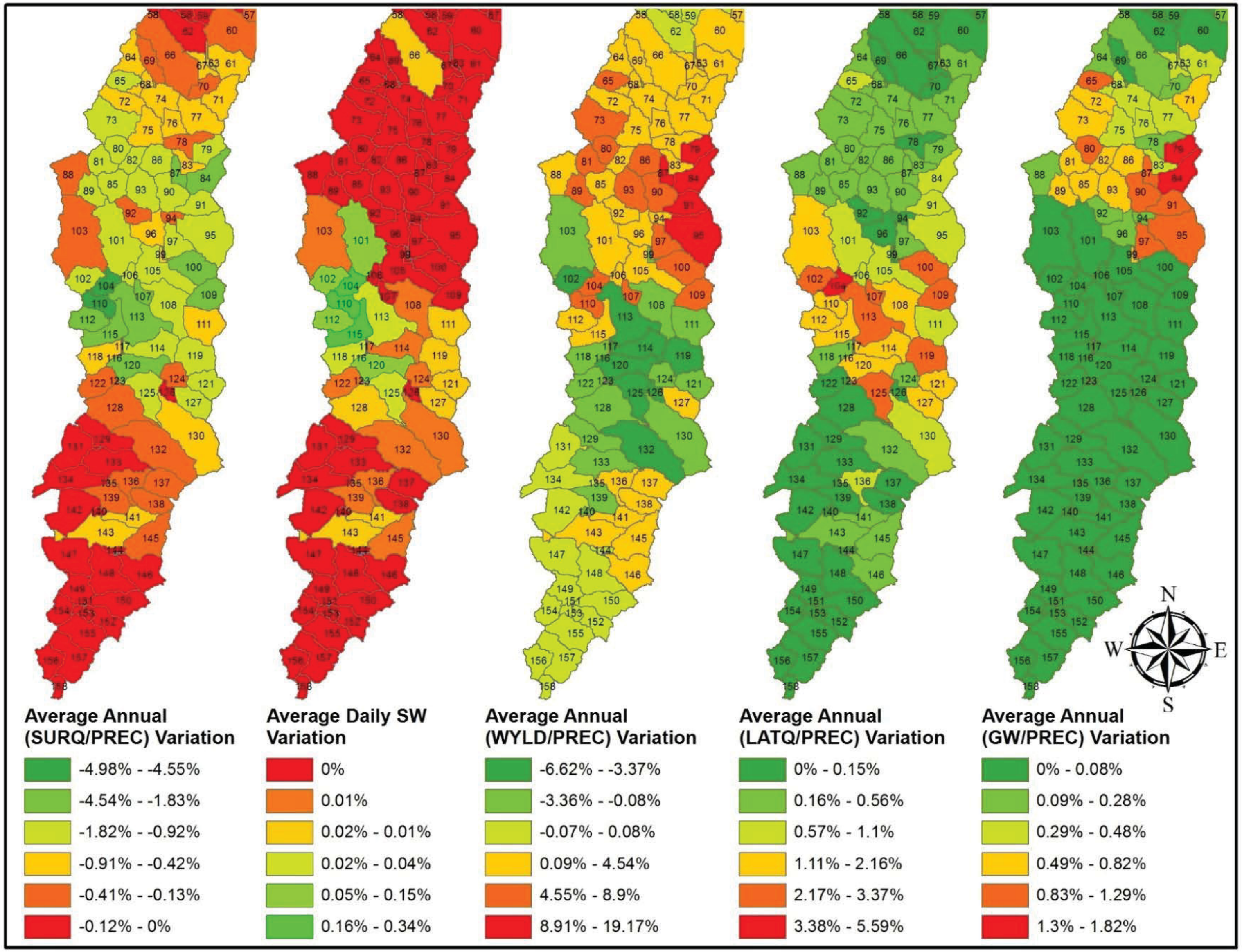


Figure 35 - Swales 70% Watershed Reductions

The swale scenario shows variations for the SURQ/PREC maximum ratio varying from -4.98% in the 10% scenario to -20.88%, in the 70% scenario which demonstrates the potential for BMPs to remove surface runoff from direct reach flow, although in a daily scale no significant effects can be noticed. Within such same scenarios, average Daily SW values varied from 0.13% to 0.64%, which in the annual scale is not representative, although increased values of SW and PERC are seen in HRU and subbasin multi-variable plot. The WYLD/PREC ratio maximum values varied from 0.37% to 6.62% in the 10% and 70% scenarios. As for LATQ, it varies from 13.22% to 3.38%. As for the GW/PREC ratio, only the subbasins around subbasin 86 showed significant GWQ increases. This is due to the region soil, which varies between LATOSSOLO and GLEISSOLO, parameterized with 8 and 2, respectively layers typically with values of SOL_K smaller than 10mm/hr.

For all variables, is noticeable the spatial relationship between swales density (Figure 21) and annual average ratio values in subbasins. However, the experienced effects in each subbasin were very varied, as they respond individually to their own topographical features, rainfall, BMP quantities and layout, and soil configuration simultaneously.

5.3.2 BMP effects: Infiltration Basins

Infiltration basins hydrological effects are dependent on soil layer depths and parameterization, as well as the HRU morphology and its parameterization, which is dependent on parameters associated with the watershed morphology (i.e.: HRU drainage length and slope). These parameters control mass transfers from SWAT land-phase reservoirs to WYLD. To depict typical BMP behaviors within this scenario, two HRU hydrographs in different subbasins (equipped with BMP devices) were chosen: HRUs 1352 and 1621.

HRU 1352 has an URBAN type of soil, which is a one-layer soil. The HRU occupies a total area of **2.079 km^2 (53.07%)** of subbasin 95 area, and its original land use is residential. The HRU was selected due to its large BMP feasibility area and as it depicts a common combination of soil and land use, besides a significant portion of land coverage within its subbasin.

In HRU 1352, lower SURQ peak values, higher Shallow Aquifer Storages (SA_ST) and Deep Aquifer Storage (DA_ST) values were observed during precipitation events. Figure 36 and Figure 37 show multi-variable plots for the 10% and 30% reduction scenarios Generally, water balances showed small Soil Water (SW) increases, with small changes in PERC and no changes in Groundwater Flow (GW_Q). Surface Runoff (SURQ) and Water Yield (WYLD), showed decreases on peaks as Lateral Flow (LAT_Q) increased simultaneously. Differences may be noticed between the 10% and 30% scenario.

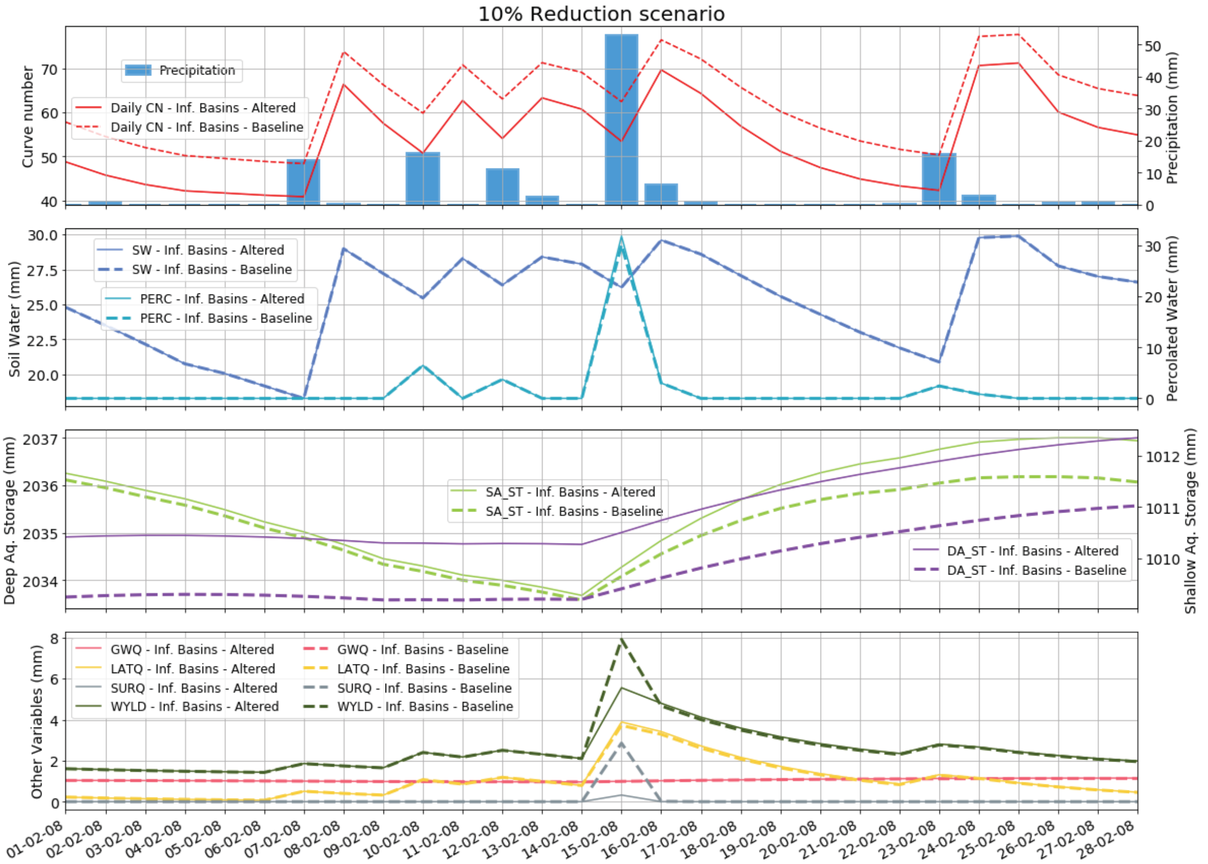


Figure 36 - 10% CN Reduction Scenario Hydrograph - HRU 1352

Differences can be noticed between the baseline scenarios, 10% reduction and 30% reduction scenarios. The 50% and 70% maintain the same values. Its graphs can be found in Annex 19 and Annex 20. Once soil saturation reaches 100% (30 mm of water on the URBAN soil reservoir), constant PERC and LATQ rates are reached, since no more water is allowed to be transferred both to aquifers, or directly to WYLD. Also, Surface Runoff reduction limit is reached, which is the reason why 50% and 70% SURQ hydrographs do not change. Groundwater storage does not vary significantly during the simulation, yielding GW_Q values close to constant.

The model calibration suggests for the TM-SQ incremental basin low LAT_TTIMEs and higher SOL_AWC than originally estimated. Under such conditions, according to the SWAT formulation, Lateral Flow (LAT_Q) is a heavy component of Water Yield (WYLD).

Also, since URBAN type soils are parameterized with a single 30 mm soil layer, only one soil layer is working as a SWAT soil reservoir. Hence, soil moisture levels are frequently above Field Capacity, and LAT_Q, DAILYCN, and WYLD become sensitive to soil moisture, responding very sharply to it, transferring water from the soil profile either as Lateral Flow or Aquifer (both shallow and deep). During larger events, Daily Surface Runoff (SURQ) is significantly reduced and LATQ increased, as soil moisture is a driving factor for LATQ. If soil storage reaches 100%, no more water can be routed.

However, despite changes in Soil Water (SW), percolated water increases are very small (~1mm) both for the Shallow Aquifer and Deep Aquifer. Another effect produced is the less sharp decreases during recession. As for the Deep Aquifer, effects long term-wise show higher storage levels than the baseline scenario, with storage levels linearly increased. Compared to HRU 1342, the swales scenario, hydrographs show larger decreases in CN, but both responses are limited to the soil saturation.

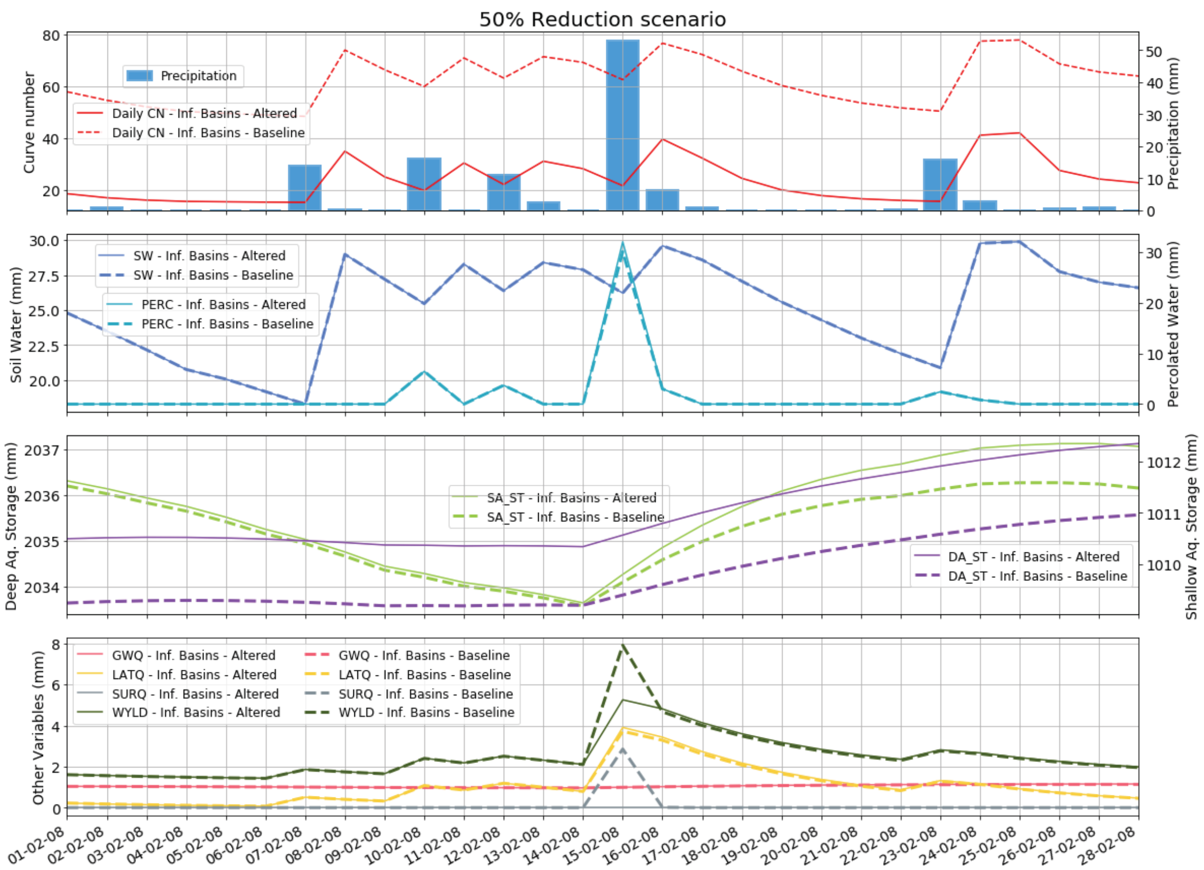


Figure 37 - 50% CN Reduction Scenario Hydrograph - HRU 1352

For Subbasin 95, the Average annual soil water increase did not assume values different than 27.10 mm during the simulation. Annual surface runoff reduced from 482 mm to 454 mm

(5.80%). Annual average Groundwater flow increased varied from 349 mm to 363 (3.8%) mm. Annual average Lateral Flow increased from 385 mm to 395(2.5%) mm. Despite the significant changes on groundwater and surface flow, water yield values did not vary. As a consequence, channel flow also did not change, despite all the compartment changes taking place in the Subbasin 95 with the addition of BMPs, or the reduced CN areas hypothesized to work as BMPs.

Changes in SURQ, and WYLD and LAT_Q can be seen in the HRU water balance, as would be expected. On the Subbasin Scale, however, water balances do not vary sufficiently to cause noticeable effects in the subbasin water yield, and hence, to the reach flow, as shown in Figure 28. Except in extreme events, where balances are significantly affected by surface runoff, no significant changes can be noticed for none of the four scenarios, as surface runoff removed is mainly distributed between lateral flow and the deep aquifer. The subbasin water balances are also controlled by soil water, as increased percolation is limited to the soil storage and once full soil saturation is reached, infiltration capacity reaches a constant value.

The effects of CN reduction for Infiltration Basin 10% and 50 % scenarios are depicted in Figure 38 and Figure 39, respectively.

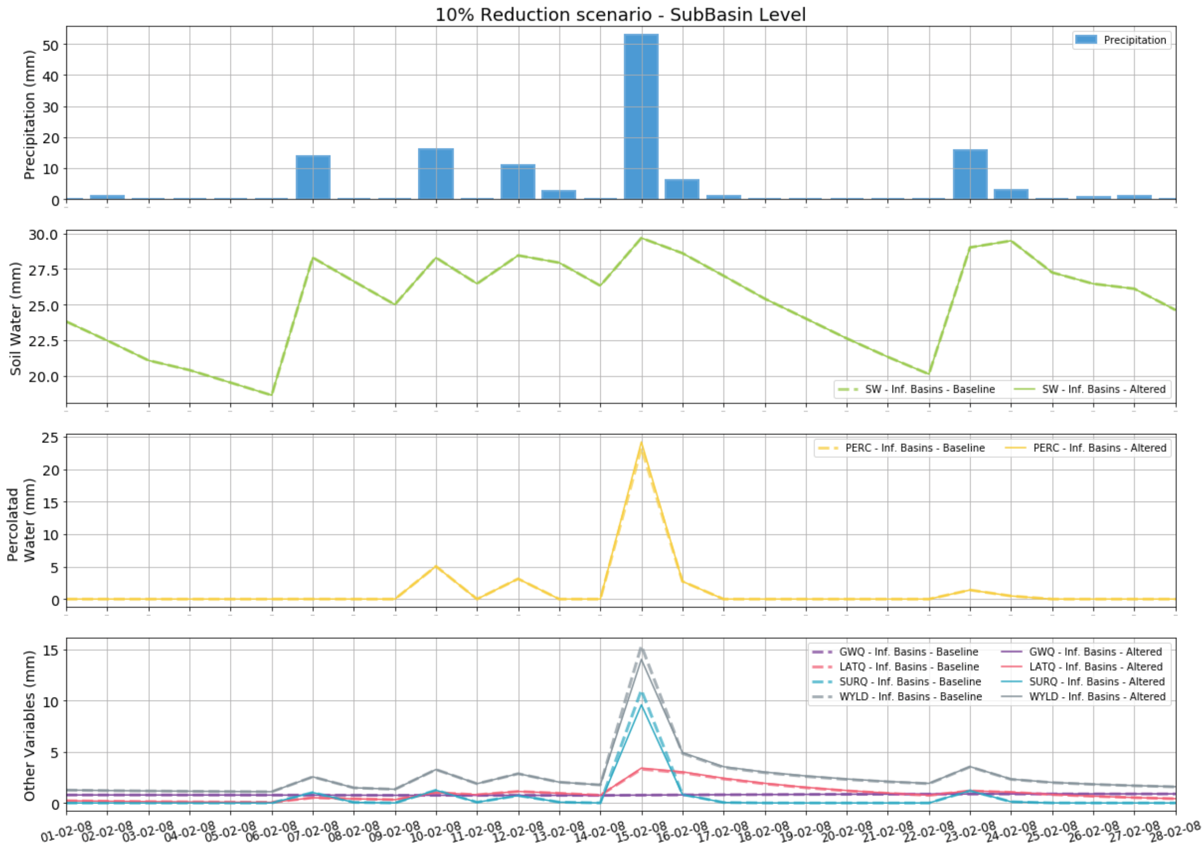


Figure 38 - 30% CN Reduction Scenario Hydrograph - Subbasin 95

Another typical Residential Land Use HRU was chosen to illustrate the effects of BMPs in non-typically soils, HRU 1621, located in Subbasin 110. The HRU occupies **1.024 Km^2 (33.32% of Subbasin area)** and is located in an ARGISOIL, which has 6 layers, with values of SOL_K lower than 100 mm/h. The six are expected to have more storage and routing capacity, as soil layers are deeper and SOL_K is larger if compared to URBAN soils. Also, increased transmission potential between soil layers and from soil to the Shallow and Deep Aquifers is expected. The calibrated value for RCHRG_DP suggests that precipitated water percolates and it is transformed in Deep Aquifer storage rather than Groundwater Flow (GWQ), and consequently lost from the system.

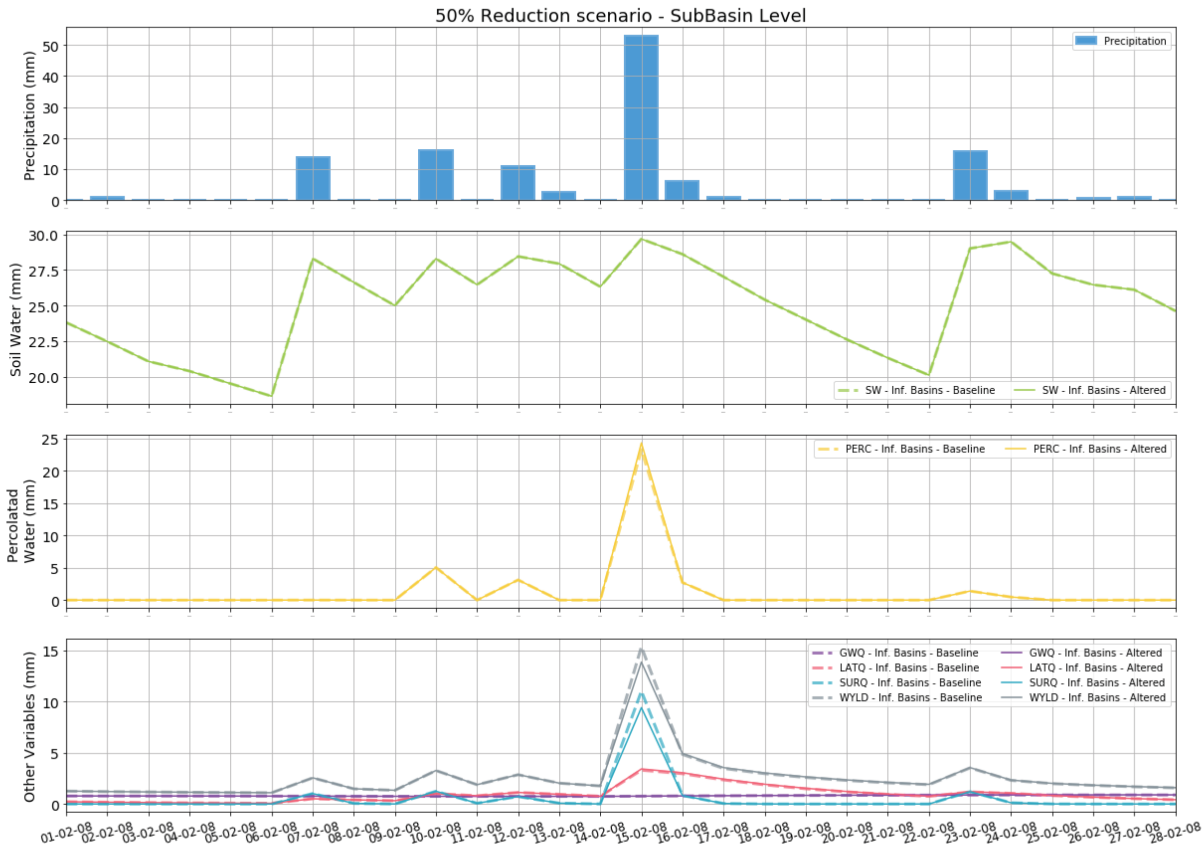


Figure 39 - 50% CN Reduction Scenario Hydrograph - Subbasin 95

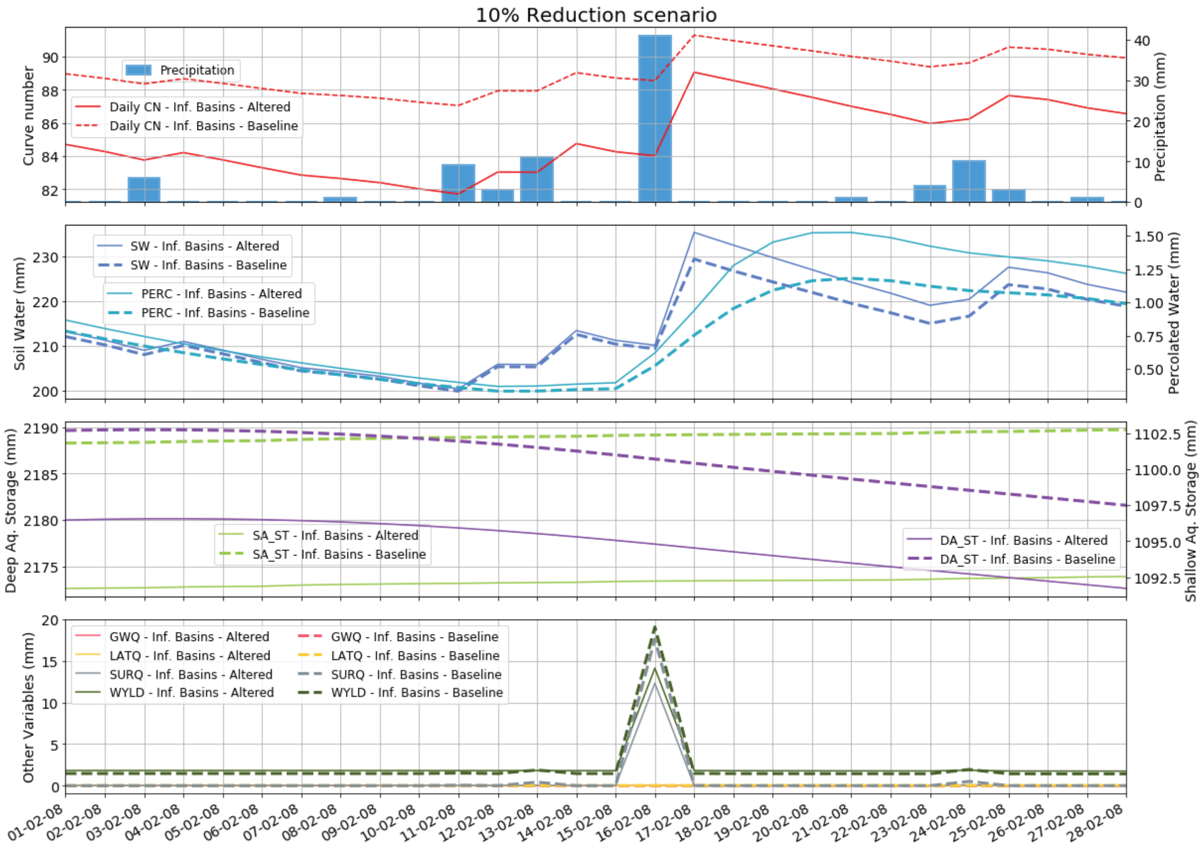


Figure 40 - 10% CN Reduction Scenario Hydrograph - HRU 1621

Figure 40 and Figure 41 depict the HRU-level reductions obtained for the 10% and 50% reduction scenarios in HRU 1554. A more reactive behavior is observed. During precipitation events, SW values increase, and CN values tend to be higher during peak events only, contrary to the observed on the Baseline scenario and in the HRU 1352 where all variables were thresholded by soil storage. PERC values increase significantly during rainfall events. Consequently, DA_ST and SA_ST increase, especially DA_ST, as the Subbasins downstream to 99(SQ-CX) are calibrated with very high RCHRG_DP values (0.943).

The changes in WYLD, LATQ and SURQ are visible for all four scenarios. Between all four scenarios, larger CN reductions yield larger WYLD baseflow values and lower event peak values. Between the 50% and the 70%, values of SURQ and LATQ are reduced to pre-event values and 100% of the surface runoff is abated and DA_ST recharge rates are increased.

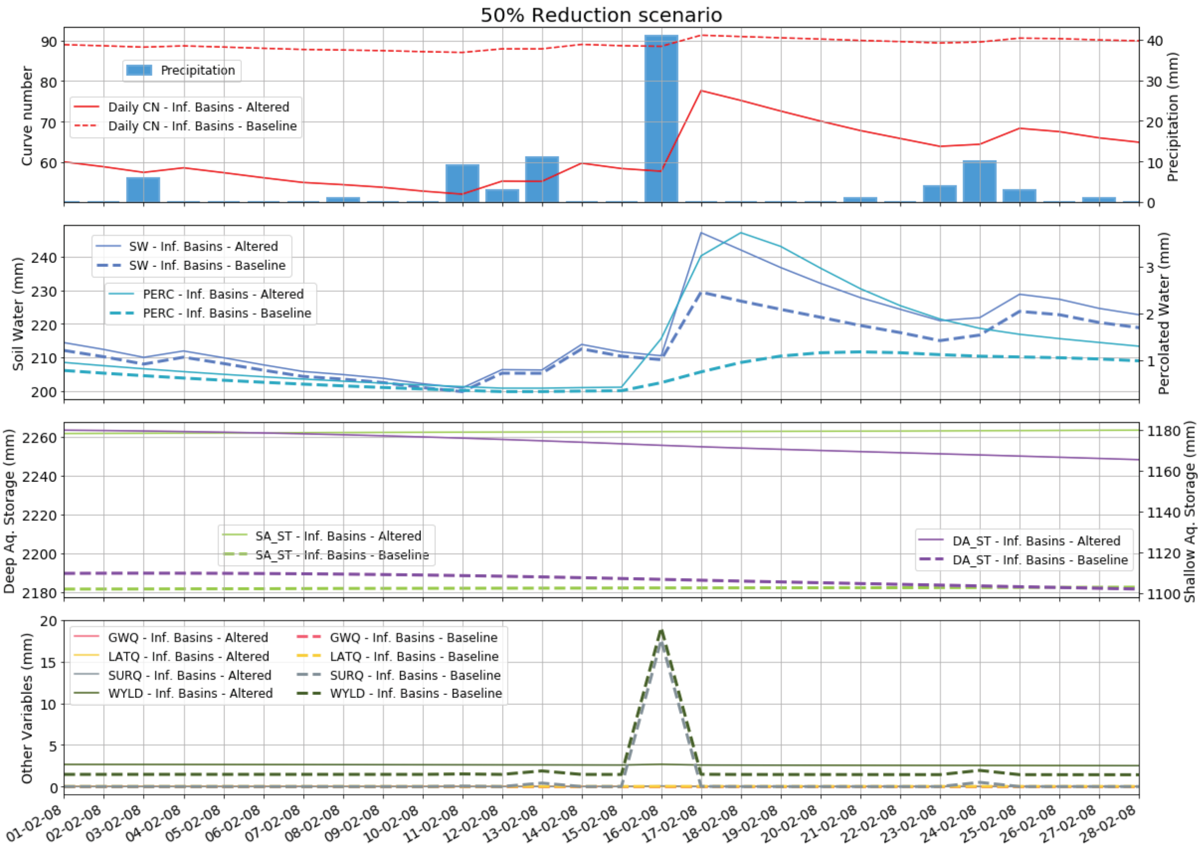


Figure 41 - 50% CN Reduction Scenario Hydrograph - HRU 1621

Although visible changes are noticeable at the HRU scale, no significant changes can be noticed at the subbasin scale. The entire subbasin response to the precipitation event is not changed by the creation of lower CN zones. In other words, WYLD component does change sufficiently for values to deviate expressively from baseline scenarios individually at the daily scale. Figure 42 and Figure 43 show the subbasin hydrographs during the month of February-2008 for Subbasin 110 under the Infiltration Basins scenario.

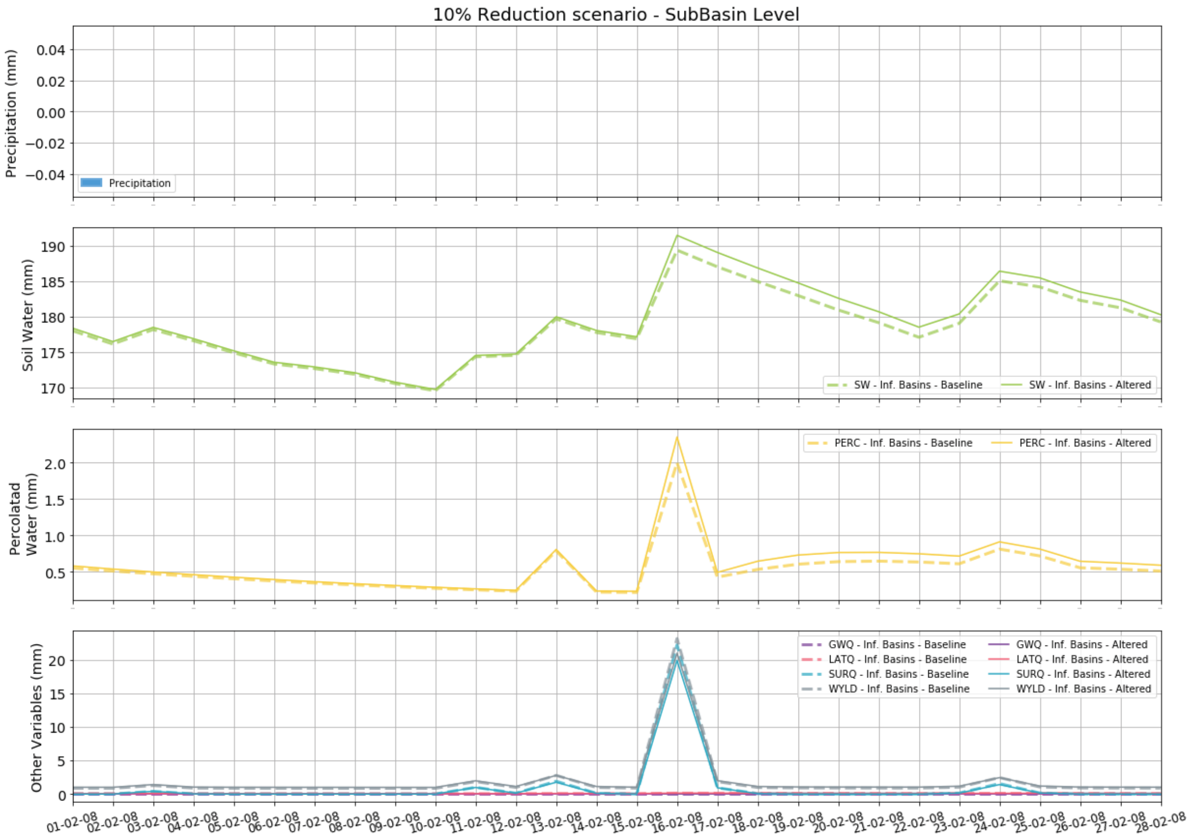


Figure 42 - 10% CN Reduction Scenario Hydrograph - Subbasin 110

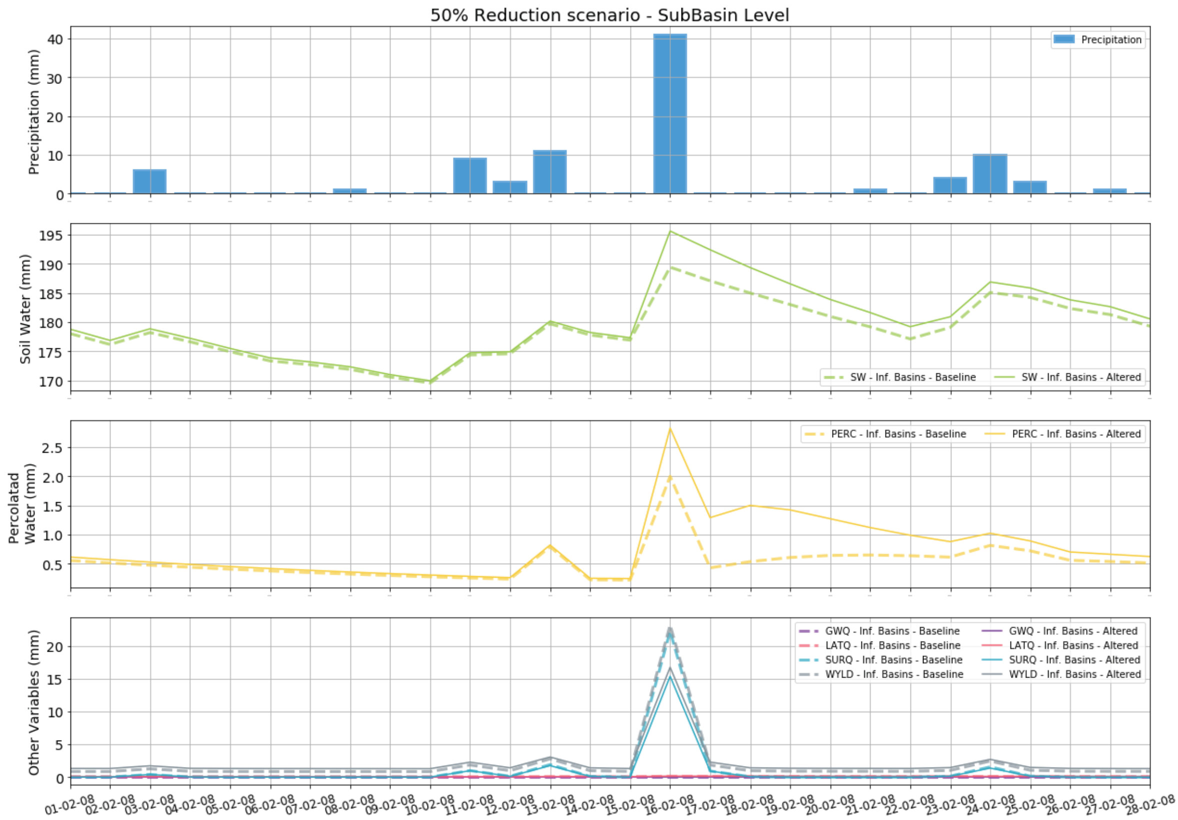


Figure 43 - 50% CN Reduction Scenario Hydrograph - Subbasin 110

The changes on CN on the subbasin level show decreases in SURQ and WYLD in the HRU scale. However, higher WYLD recession values compute higher WYLD/PREC ratios and lower SURQ/PREC values. This means that Subbasin 95 daily peaks are abated, but the higher baseflow values contribute to higher annual average flows. Figure 44 and Figure 45 depict in the subbasin scale the annual mean reductions during the Simulation Years (200-2014), not counting the warm-up period.

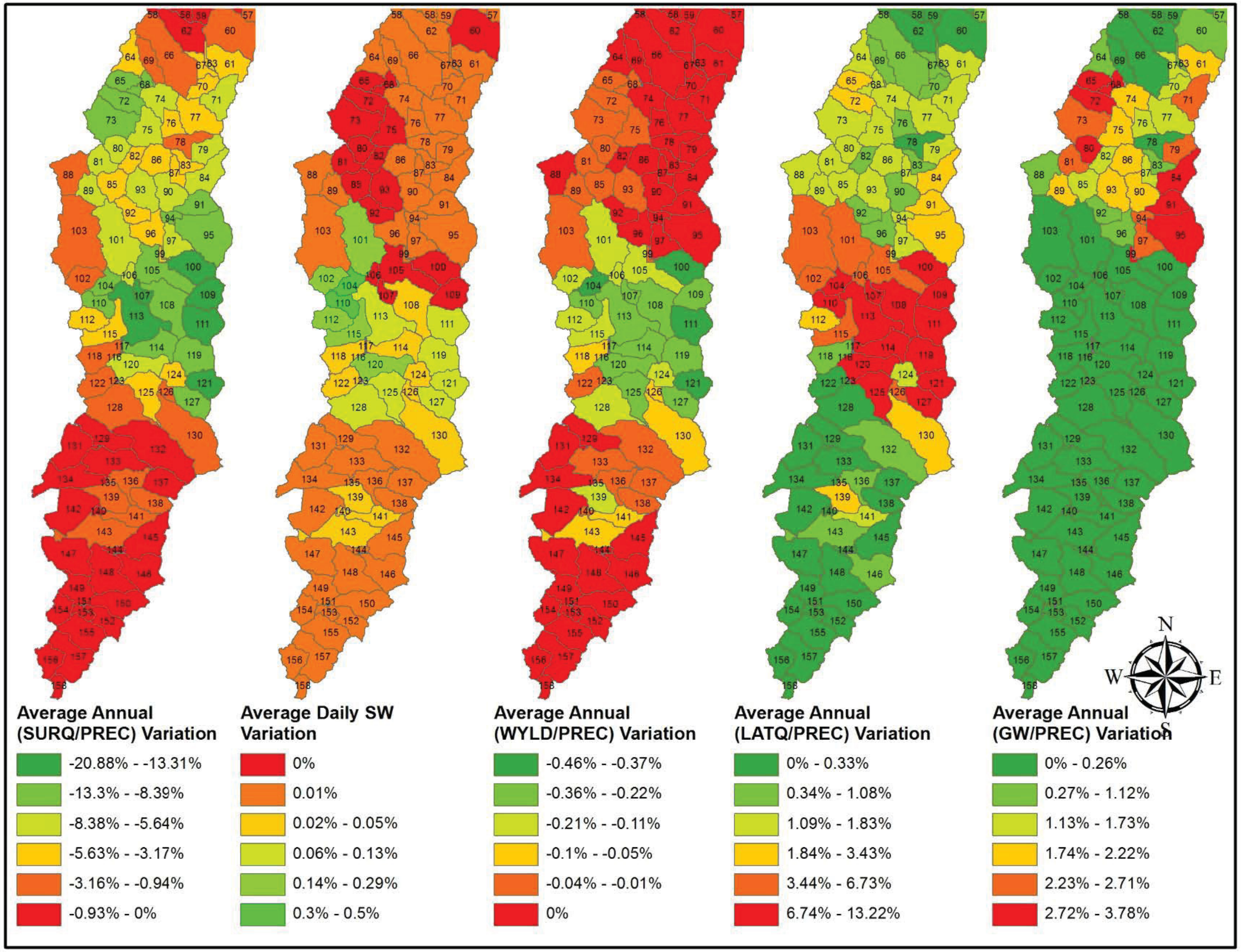


Figure 44 - Infiltration Basins 10% Watershed Reductions

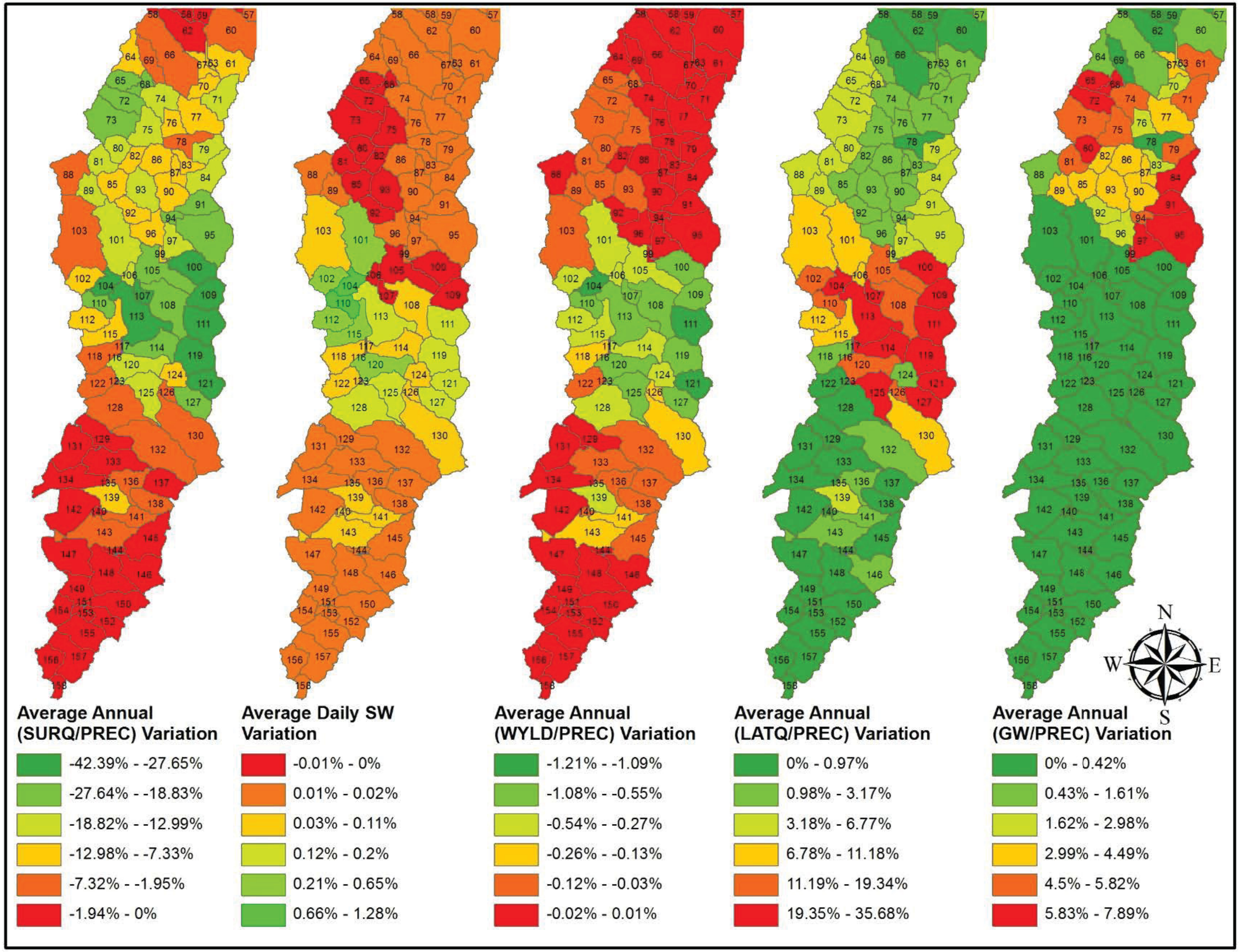


Figure 45 - Infiltration Basins 70% Watershed Reductions

The figure depicts that SURQ/PREC typically decreases in all subbasins. Annual WYLD/PREC, shows a thresholded behavior, as reservoirs in HRUs become saturated, and the exceeding water is routed back to SURQ. LATQ shows increases up to 8.67%, despite many subbasins depict no changes. GW/PREC contributions are not typically followed by any other variable (SEE SW LATER). GWQ/PREC behavior is variable between different CN scenarios, as seen in Subbasins 89 or 97, where different results are obtained from various CN values. As Soil reservoirs are more or less close to full saturation, water may or may not be routed, or routed at larger ratios, which makes this variable tracking very difficult, in order to quantify the GWQ/PREC.

5.3.3 BMP effects: Biorretention

Biorretention effects Field-scale (HRU) and Subbasin scale are dependent on soil layer depths and parameterization. To depict typical BMP behaviors, two HRU hydrographs, in different subbasins, equipped with BMP devices were selected: HRUs **1248** and **1480**.

Figure 46 and Figure 47 depict the behavior in HRU number 1248 for the 10% and 50% reduction scenarios. HRU 1248 is located in subbasin 95. The HRU is parameterized with an URBAN type of soil, which is a one-layer horizontal profile. The HRU occupies a total area of 0.991 km^2 (25.30%) of the subbasin area, and its original land use is residential. The area was chosen due to its large BMP feasibility area as it depicts a common combination of soil and land use, besides a significant portion of land coverage within its subbasin.

In the HRU scale, lower SURQ peak values, higher Shallow Aquifer Storages (SA_ST) and Deep Aquifer Storage (DA_ST) values were observed. In general, water balances show small increases in Soil Water (SW), for the 10 and 30% scenarios and no increases in Groundwater Flow (GW_Q). As for Surface Runoff (SURQ) and Water Yield (WYLD), the first decreases on peaks and the second show reduction on the 10 and 30% scenarios. The cause is variation in soil saturation (SW), which increases CN values during the recession period, stopping more water from being routed to soil and groundwater reservoirs. Lateral Flow (LAT_Q) shows slight increase during peaks and recession. The effects of CN reduction for Biorretention 10% and 50% CN reduction scenarios is depicted in Figure 46 and Figure 47.

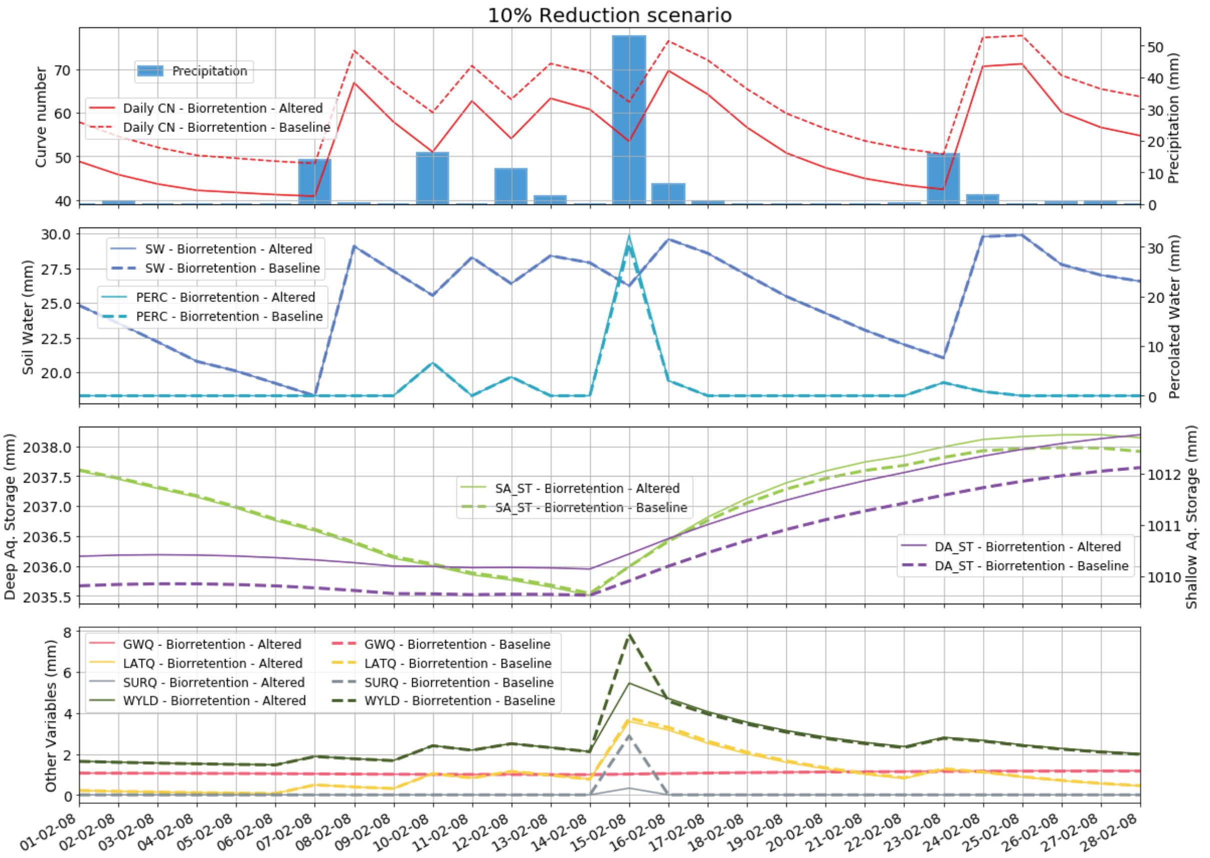


Figure 46 - 10% CN Reduction Scenario Hydrograph - HRU 1248

The baseline scenarios, 10% reduction and 30% reduction scenarios show visible differences from baseline scenarios on water balances. Such difference remains constant both for the 50% reduction and the 70% reduction scenarios. Its Graphs can be found in Annex 0 and 0. The model calibration suggests for the TM-SQ incremental basin low Lateral Travel Times (LAT_TTIME, ranges from 0-20) and high Soil Available Water Content (SOL_AWC) than originally estimated. As these two variables are calibrated to be relevant for the HRU WYLD, according to the SWAT formulation, Lateral Flow (LAT_Q) is coerced to be the driving force for Water Yield (WYLD) values. Also, since URBAN type soils are parameterized with a single 30mm soil layer, only one “SWAT soil reservoir” routes precipitation. As a consequence, soil moisture levels are frequently at or above Field Capacity moisture levels, and LAT_Q, DAILYCN, and WYLD become sensitive, responding very sharply to it, as water from the soil profile is routed either as Lateral Flow or Aquifer (both shallow and deep). During larger events, Daily Surface Runoff (SURQ) is significantly reduced and LATQ increased, as soil moisture is a driving factor for LATQ. However, despite changes in Soil Water (SW), percolated water increases are very small (~1mm) both for the Shallow Aquifer and Deep Aquifer. In Shallow aquifer storage, particularly, the effect produced is of less acute decreases during recession. Deep Aquifer long term effects showed higher storage levels than the baseline scenario, being directly positive-shifted during the BMP simulation. As soil moisture controls directly the HRU Water Yield (WYLD) and the water table does not change significantly in storage, GW_Q changes does not change significantly during simulations.

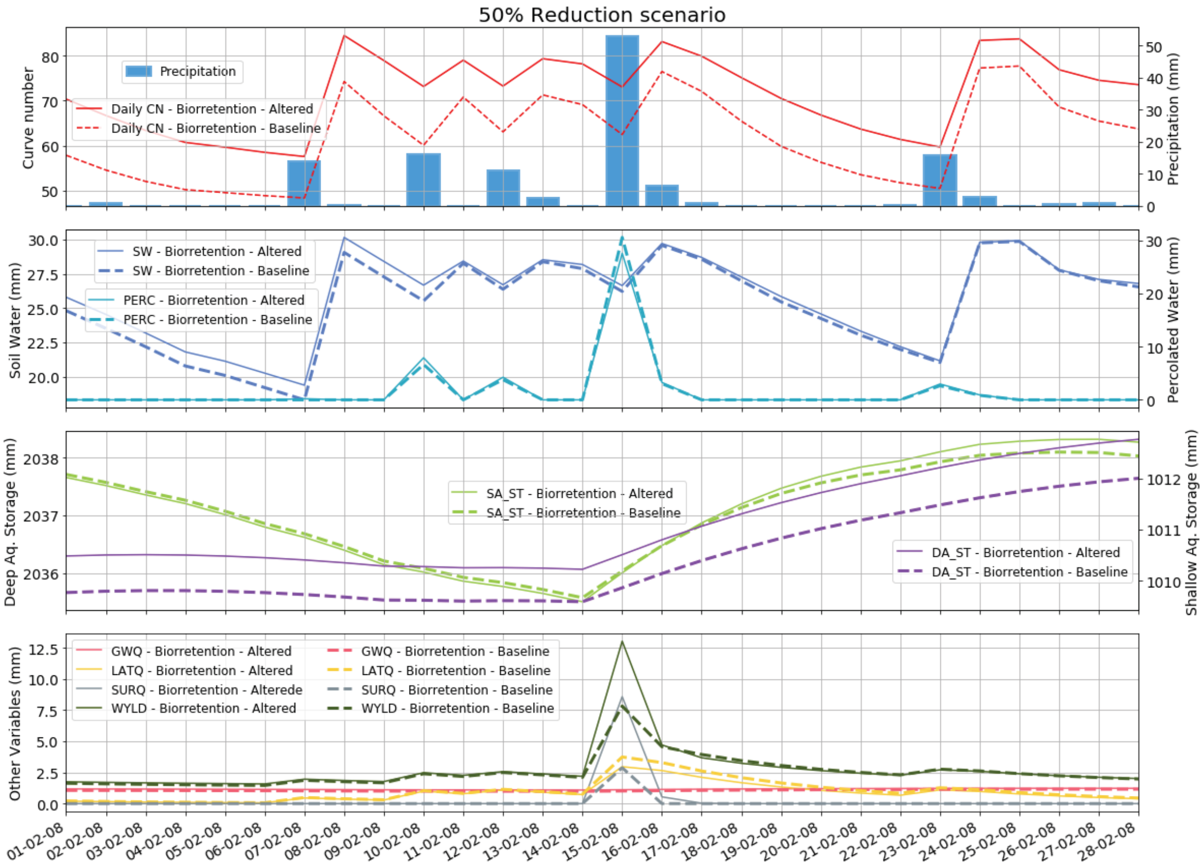


Figure 47 -50% CN Reduction Scenario Hydrograph – HRU 1248

Changes are visible in the HRU-scale water balance, as it would be expected for a typical BMP watershed. On the Subbasin Scale, however, water balances do not change sufficiently to cause noticeable effects in the water yield on the subbasin, and hence, to the reach flow, as shown in Figure 48 and Figure 49. Even in extreme events, where balances are significantly affected by antecedent moisture conditions and groundwater recharge, no significant changes on WYLD or its components can be noticed for neither of the four scenarios.

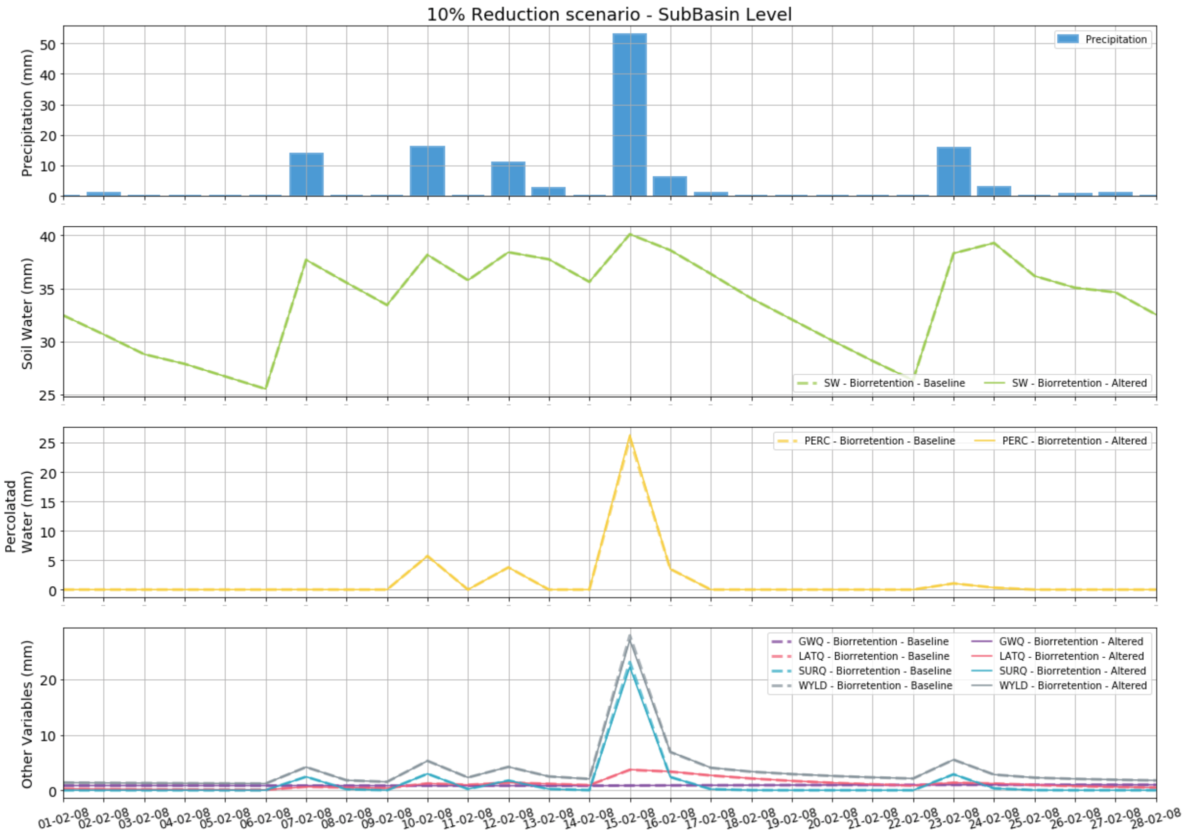


Figure 48 - 10% CN Reduction Scenario Hydrograph - Subbasin 95

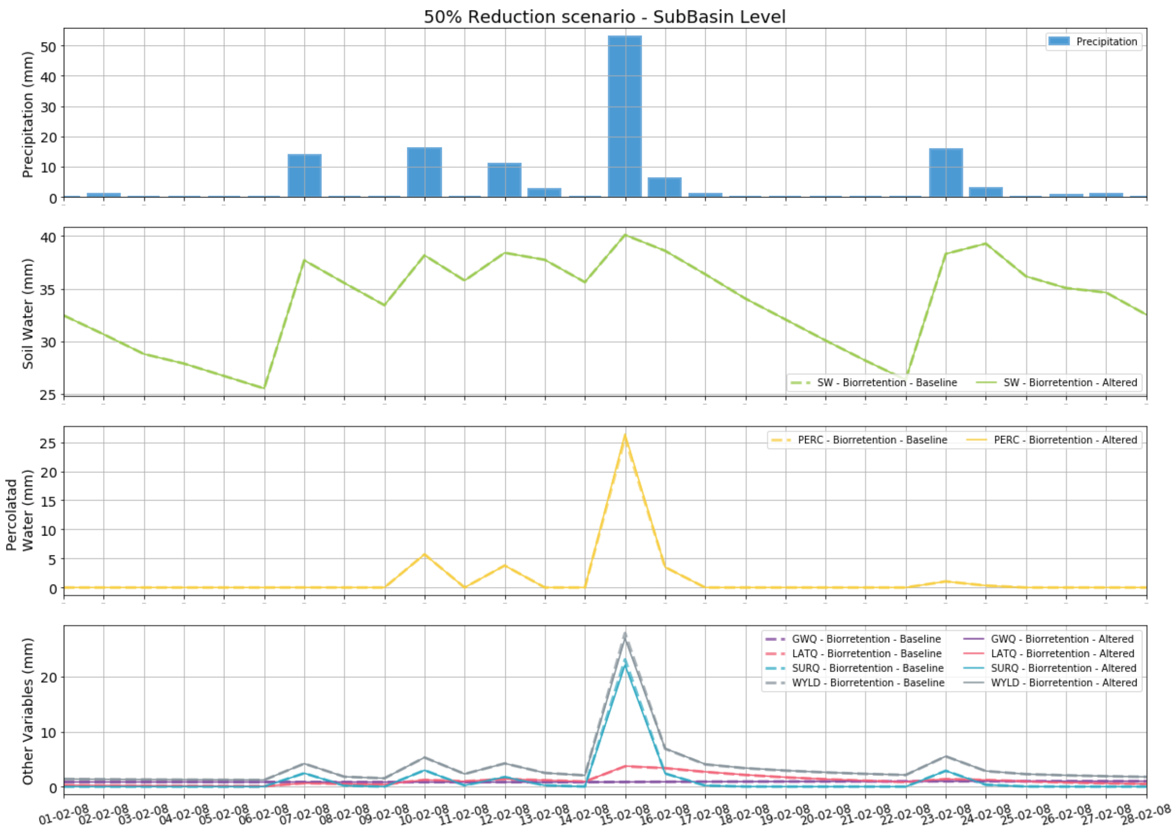


Figure 49 - 50% CN Reduction Scenario Hydrograph - Subbasin 95

In Subbasin 95, for the 70% scenario, average annual soil water remained in 27,1mm. Annual surface runoff changing from 482 mm to 455 mm (5,72%). Groundwater flow annually varied from 349 mm to 363 mm (3.85%). Lateral Flow increased from 386 mm to 394 mm (2.07%). Despite the significant changes on groundwater and surface flow, water yield values did not vary. As a consequence, changes were also not observed in channel flow, despite all the compartment changes taking place in the subbasin with the addition of BMPs, or the reduced CN areas hypothesized to work as BMPs.

Another typical Residential Land Use HRU was chosen to illustrate the effects of BMPs in non-typically urban soils, HRU 1480, located in Subbasin 110. The HRU occupies an area of **0.235 km² (22.94% of Subbasin area)** and is has an ARGISOIL soil type, a 6-layer horizontal profile, typically with values of SOL_K lower than 100mm/h. The calibrated value for RCHRG_DP in the TM-SQ incremental basin suggests that precipitated water percolates and it is transformed in Groundwater flow (GW_Q) rather than Lateral Flow (LAT_Q). Hence, groundwater storage and recession changes are expected.

Figure 50 and Figure 51 show the HRU-level reductions obtained for the 10% and 50% reduction scenarios in HRU 1480. In both graphs a nonlinear response to CN variation is observed, when compared to HRU 1248. As more water is infiltrated, SW values and CN values tend to increase more abruptly during peak. PERC values increase significantly and show longer recession periods during and after rainfall events. Consequently, DA_ST and SA_ST increase, especially DA_ST, as the Subbasins downstream to 99(SQ-CX) are calibrated with very high Shallow and Deep Aquifer partitioning values.

The changes in WYLD, LATQ and SURQ are visible for all four scenarios. Among scenarios, larger CN reductions yield smaller WYLD baseflow values and lower event peak values. Between the 50% and the 70%, values of SURQ and LATQ are reduced to pre-event values and 100% of the surface runoff is abated and DA_ST recharge rates are increased.

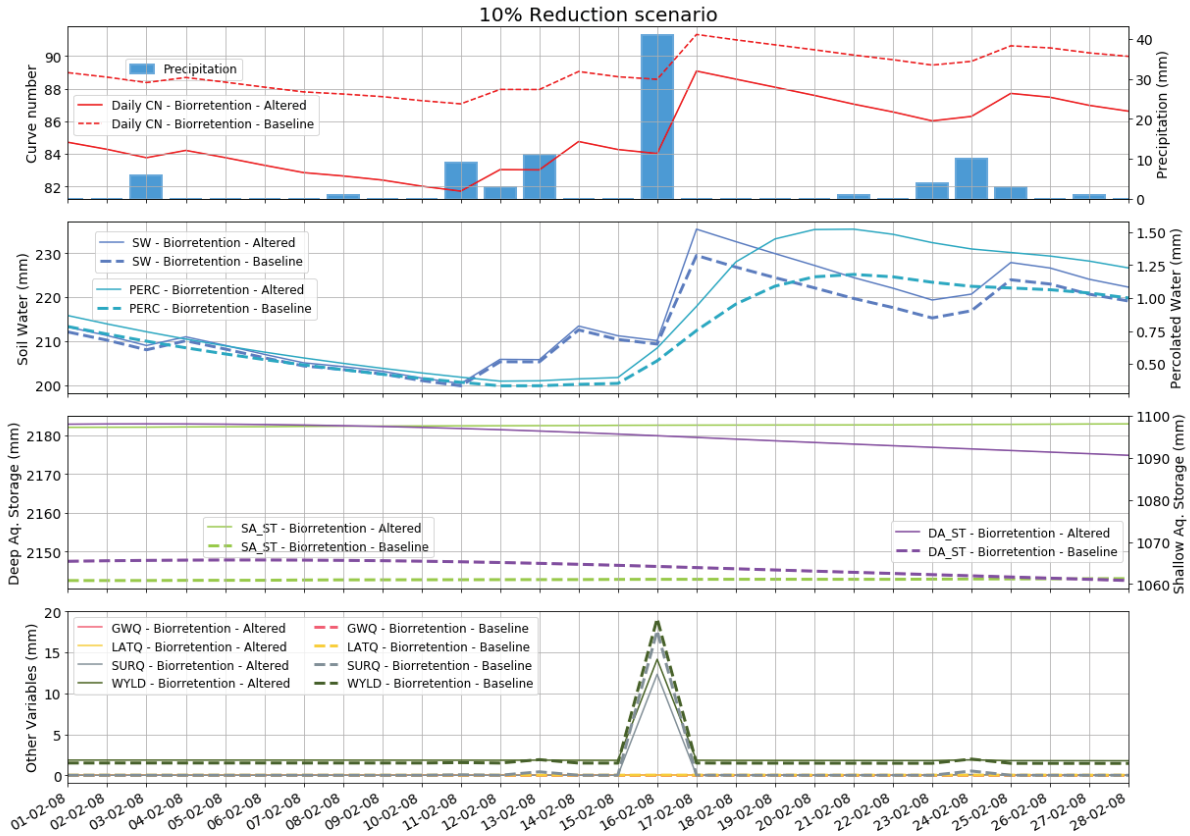


Figure 50 - 10% CN Reduction Scenario Hydrograph – HRU 1480

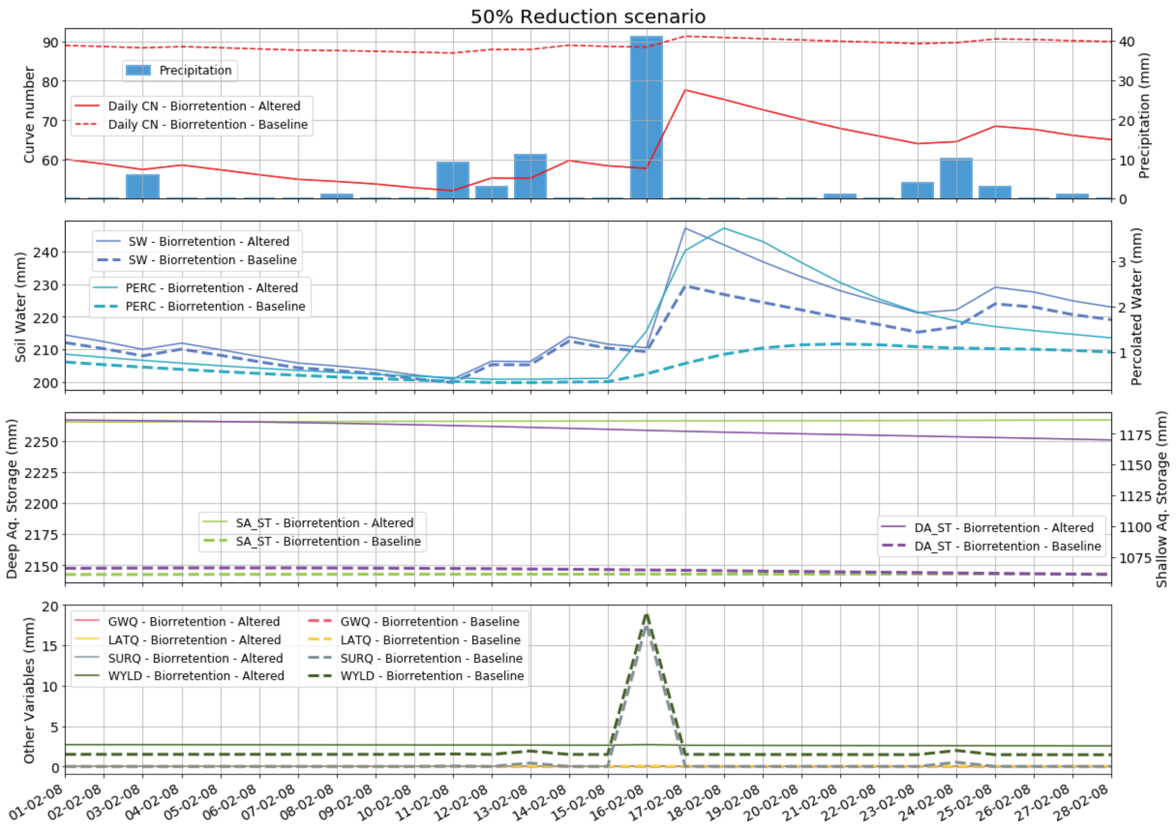


Figure 51 - 50% CN Reduction Scenario Hydrograph – HRU 1480

In Subbasin 110, for the 70% Infiltration Basins scenario, average annual soil water increased from 221 to 223mm (0.89%). Annual surface runoff decreased from 930 mm to 455 mm (51,07%). Groundwater flow did not vary during the simulations. Lateral Flow increased from 52 mm to 56 mm (7.7%). The HRU hydrographs show decreases in SURQ and WYLD. The subbasin is retaining more water, differently from HRU 1248, where SURQ was predominantly routed as LATQ.

The changes on CN at the subbasin level showed decreases in SURQ and WYLD, in the HRU scale. However, higher WYLD recession values compute higher WYLD/PREC ratios and lower SURQ/PREC values. This means that subbasin 95 daily peaks are abated, but the higher baseflow values contribute to higher annual average flows. Figure 52 and Figure 53 depict in the subbasin scale the annual mean reductions during the Simulation Years (200-2014), not counting the warm-up period.

Although visible changes are noticeable at the HRU scale, changes can only be noticed in SW and PERC values, which confirm the hypothesis of SURQ being stored in the subbasin. Both values increase in peaks and have their recession rates changed by the CN values. However, WYLD and none of its components did vary significantly with the creation of reduced CN zones. Consequently, no changes can be seen in the subbasin reach hydrographs. Figure 52 and Figure 53 depict Subbasin 110 hydrographs for the period of 01-feb-2008 to 28-feb-2008.

Figure 52 - 10% CN Reduction Scenario Hydrograph – Subbasin 110

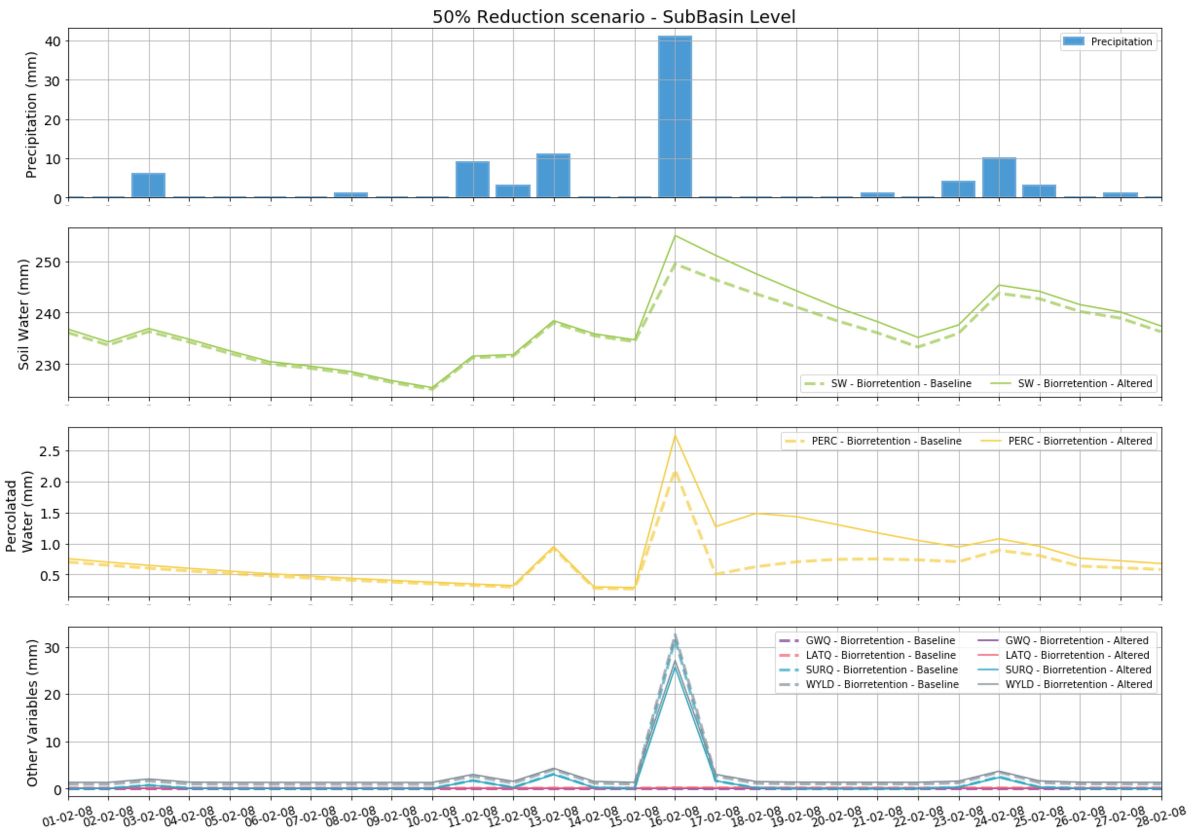


Figure 53 - 50% CN Reduction Scenario Hydrograph – Subbasin 110

Figure 54 and Figure 55 depict on subbasin scale the annual mean variations for Biorretention 30% and 70% reductions during the Simulation Years (2006-2014), not counting the warm-up period. The 10% and 50% scenarios can be found in Annex 0 and 0. In the subbasin scale during the entire simulation, changes can be seen in the SURQ/PREC ratio, as is consistently decreases in all four scenarios.

For all variables, no significant association between the Biorretention density map and variable behavior can be perceived. The CN reductions and BMP reparameterization does not affect subbasins between 1-95 for most parameters. The experienced effects in each subbasin is very varied, as they respond individually to their own topographical features, rainfall, BMP quantities and layout, and soil configuration simultaneously.

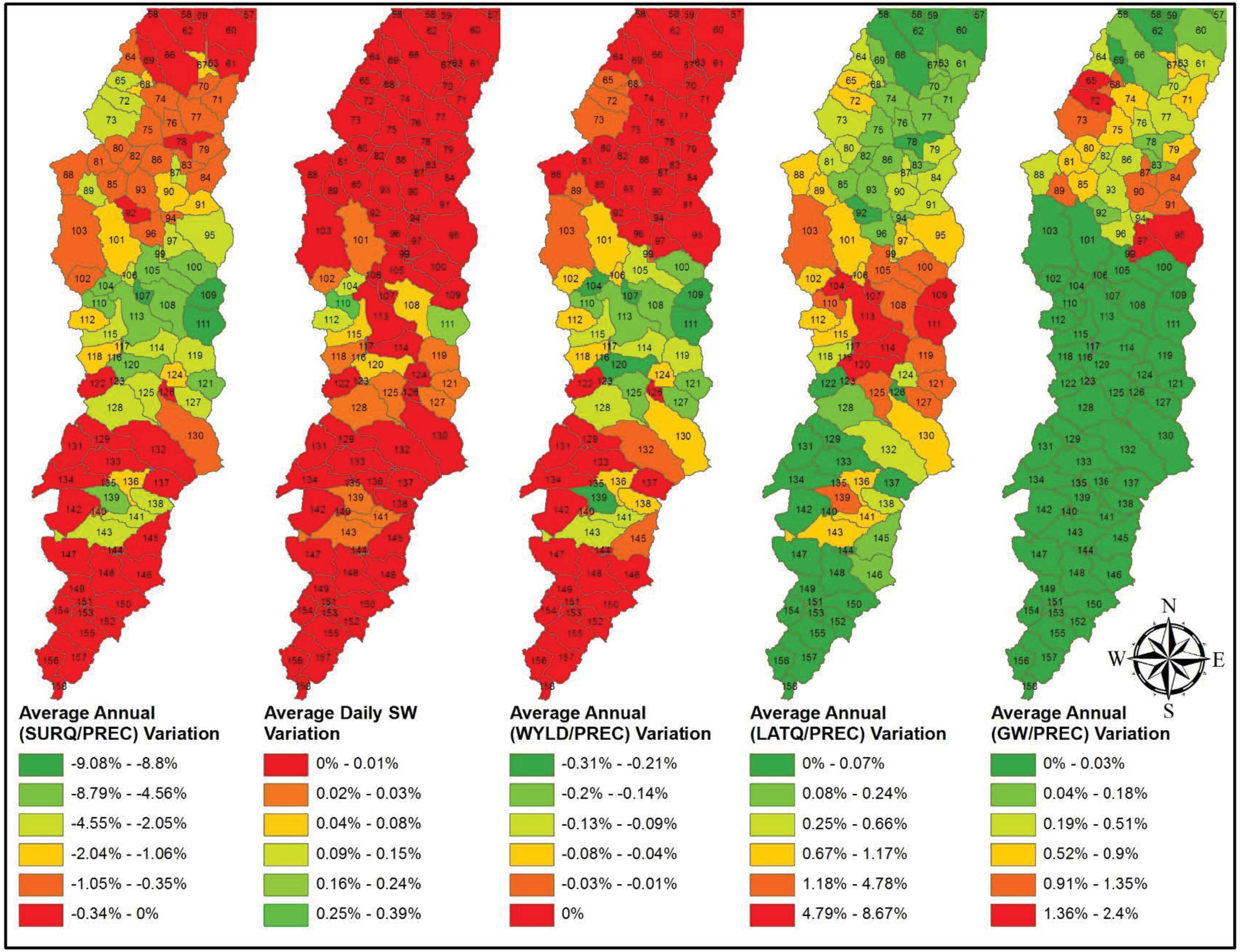


Figure 54 - Biorretention 10% Watershed Reductions

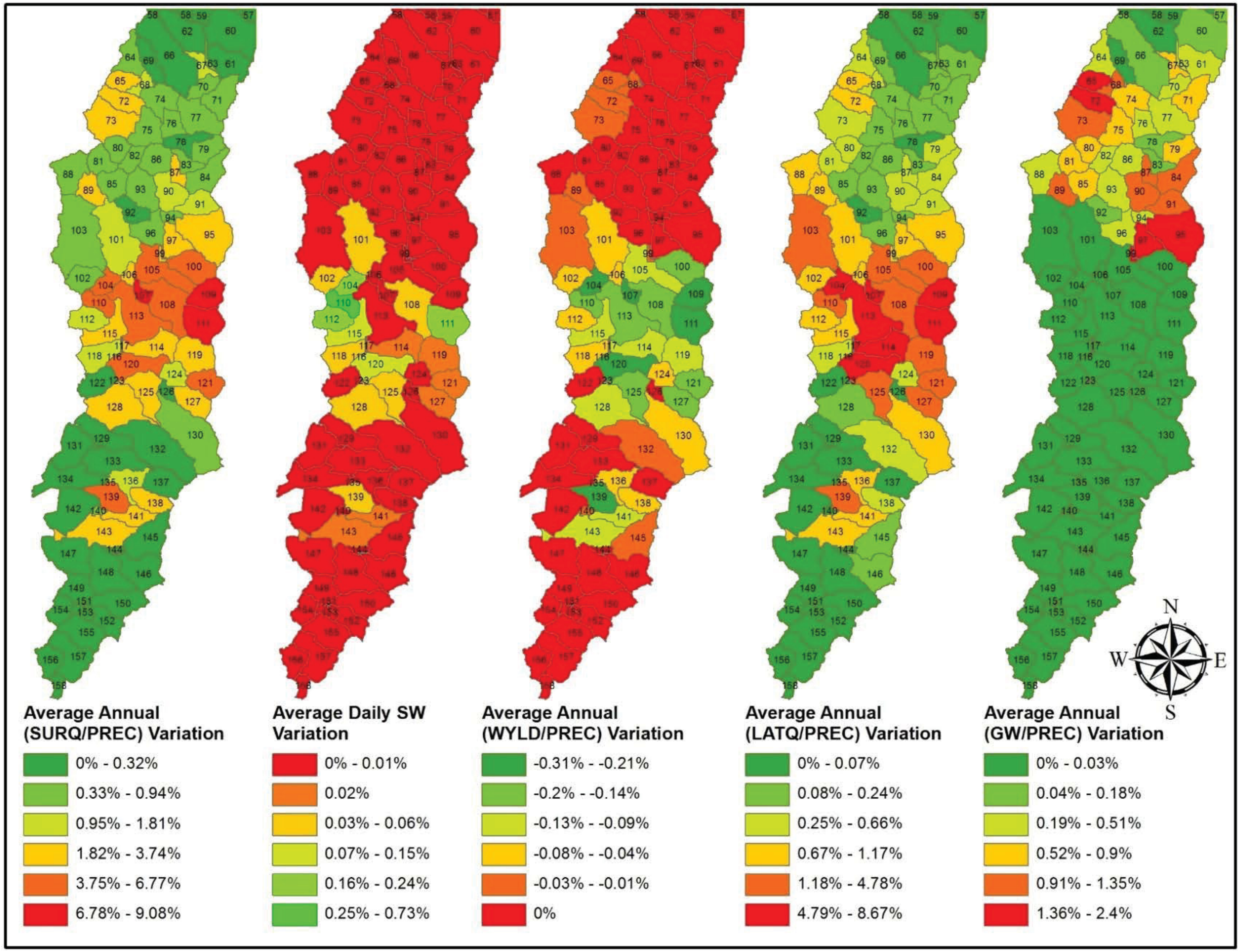


Figure 55 - Biorretention 70% Watershed Reductions

The average Reach Flow (FLOW_OUT_cms) during the simulation for the reaches 99 and 155(SQ and TM, respectively) is shown in Figure 56 to Figure 61. As seen from results, daily peaks are not reduced, especially on extreme events, as the few subbasins export lower quantities of WYLD, and others, such as subbasin 95 – see Figure 35, those quantities are not sufficient to provoke WYLD changes visible at the Subbasin Scale.

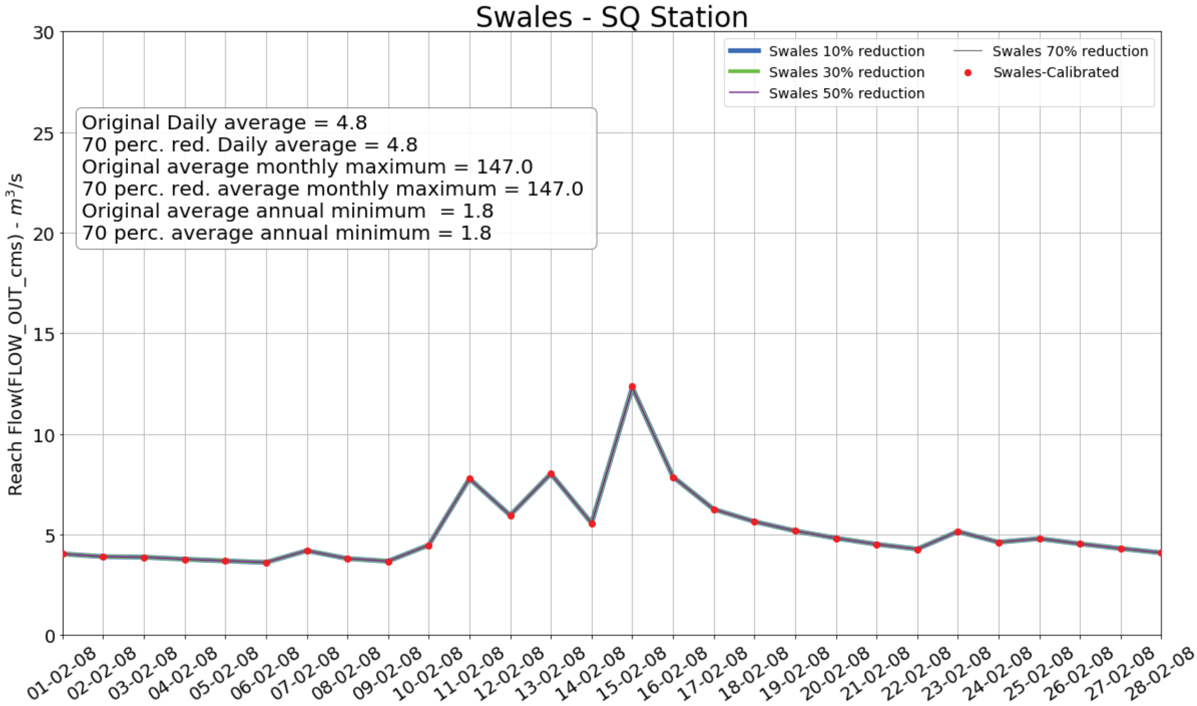


Figure 56- Reach Flow - Swales scenario - SQ Station

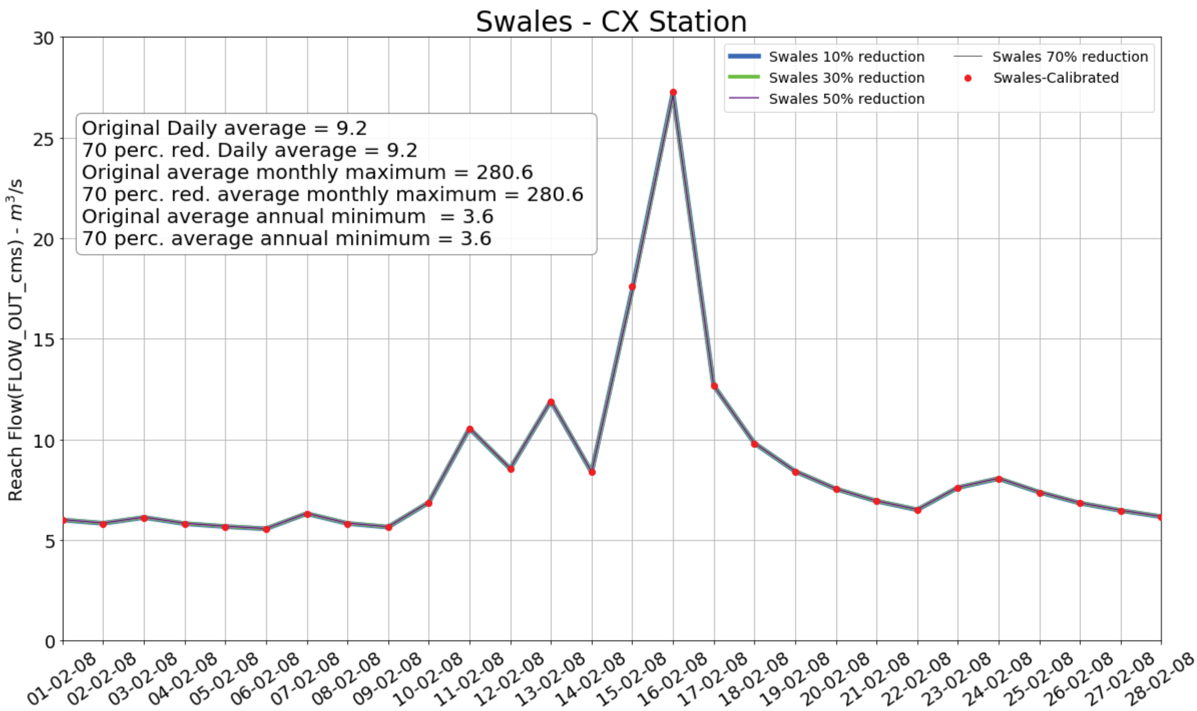


Figure 57 - Reach Flow - Swales scenario - CX Station

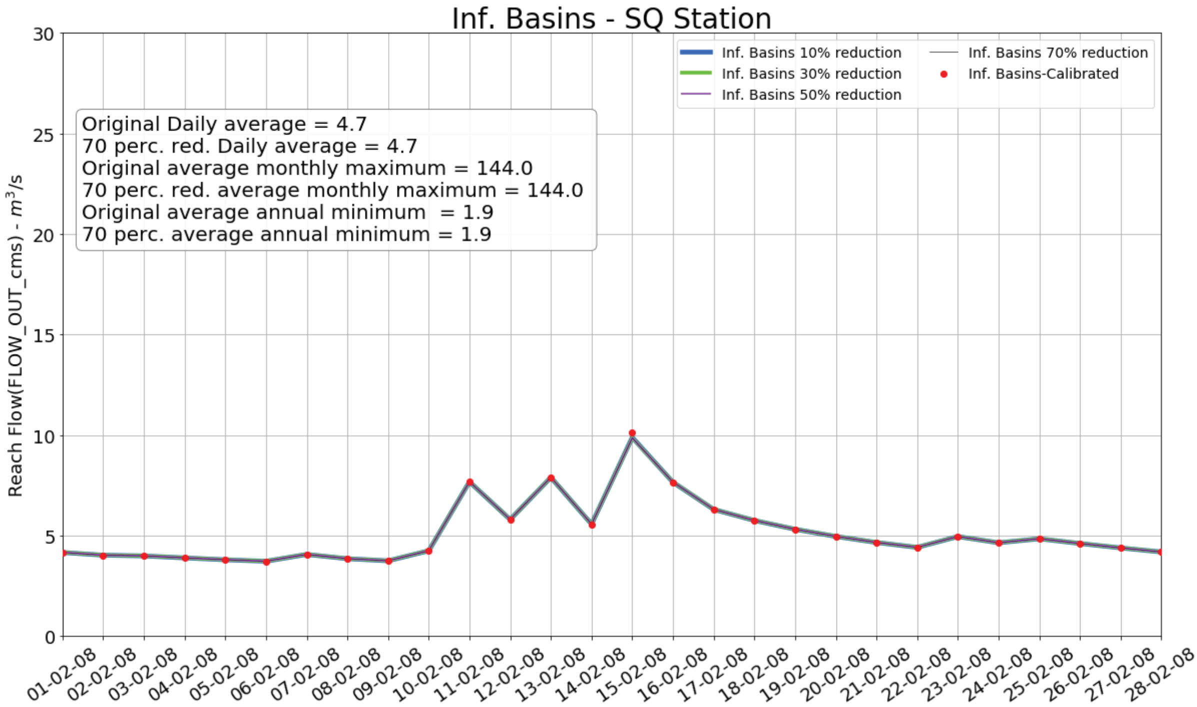


Figure 58 - Reach Flow – Infiltration Basins scenario - SQ Station

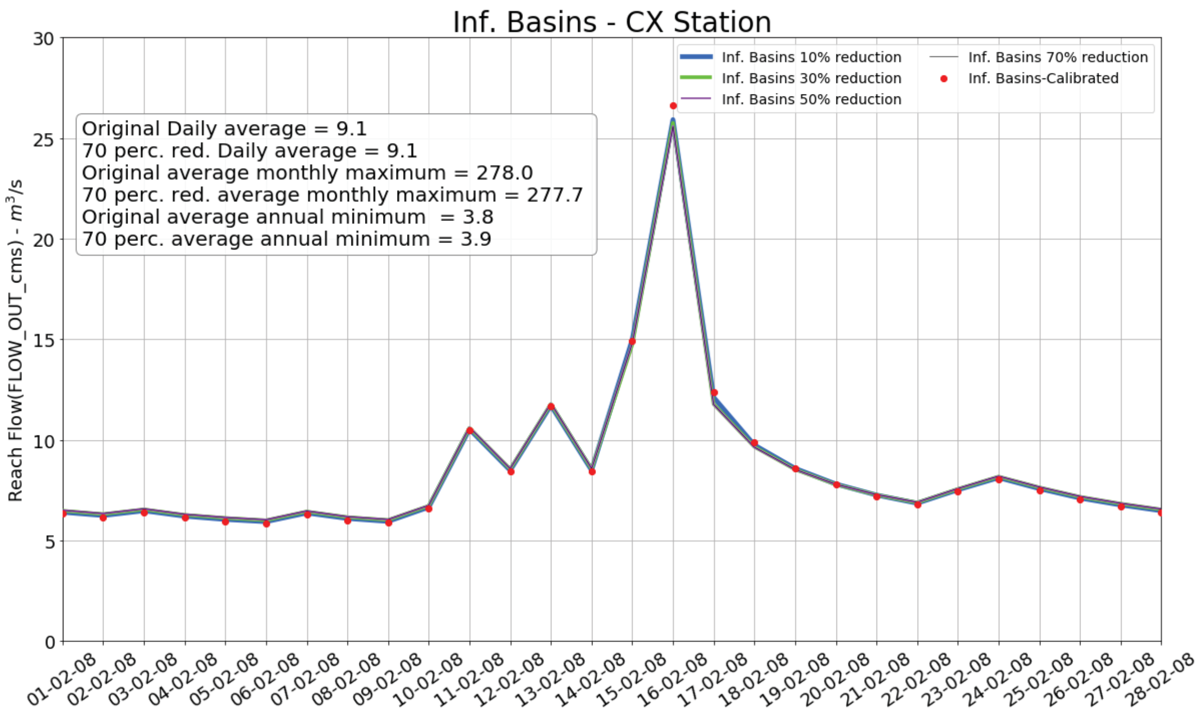


Figure 59 - Reach Flow – Infiltration Basins scenario - CX Station

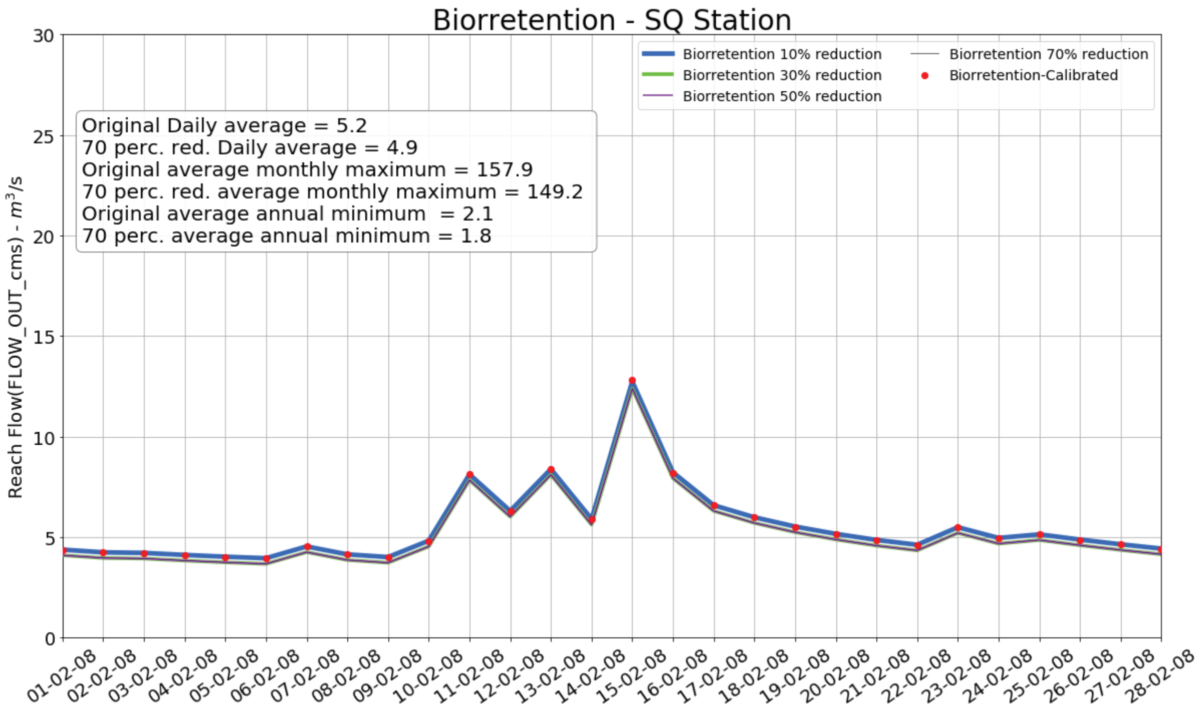


Figure 60 - Reach Flow – Biorretention scenario - SQ Station

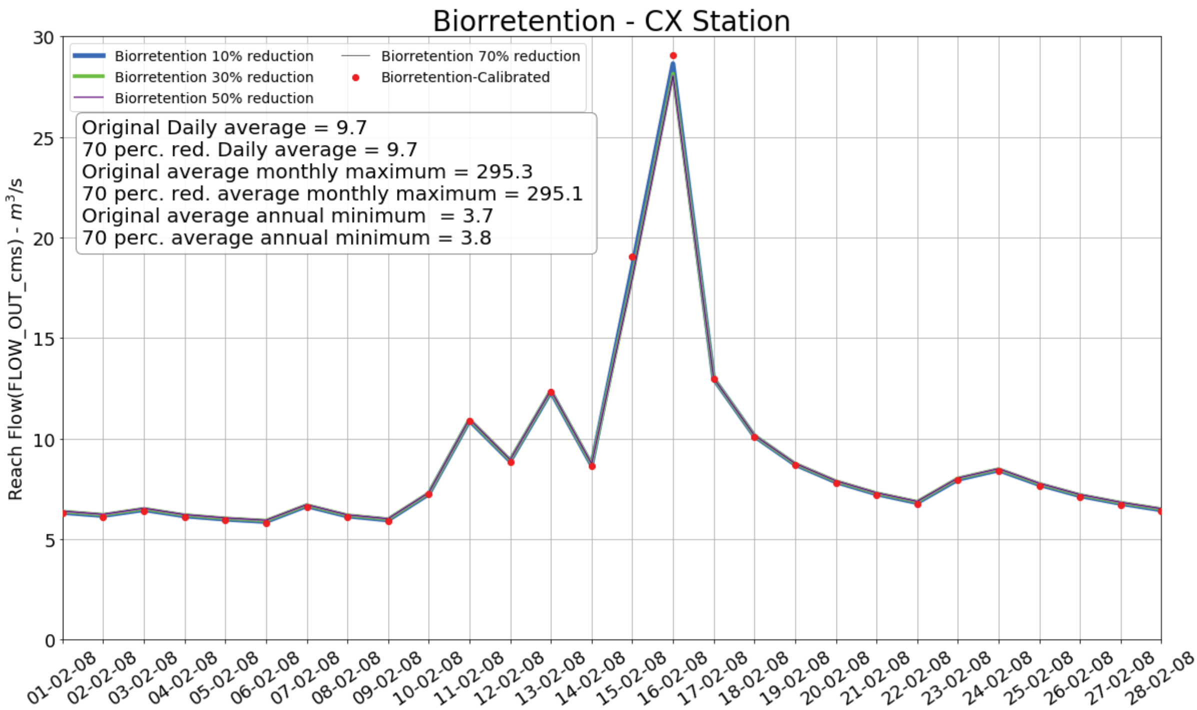


Figure 61 - Reach Flow – Biorretention scenario - CX Station

The differences between the baseline and reduced scenarios are very small, with correlation coefficients close to 1 for all simulation scenarios. The only BMP that showed some difference is the Biorretention device. Biorretention devices, as showed above, did not demonstrate significant changes for all parameters at the **HRU, subbasin and Watershed**

Annual scale. However, the large-area and distribution caused noticeable changes to reach flow in SQ station. The trend is not followed by the result obtained for the CX station.

5.3.4 BMP effects: Overall conclusions

According to SWAT formulation, hydrological effects are dependent on soil parameterization, HRU slope, and HRU Drainage length. These values, upon calibration, define subbasin main mass transfer rates and limits. For simplicity two HRUs were examined in each CN reduction scenario, as a first approach to the obtained data. The techniques used in this work (placement + reparameterization) force the assessment of all HRUs during all timesteps to thoroughly assess the occurrence and intensity of the reductions. However, evaluating such large datasets is not a trivial task, and more effort would be required to extract the results to their full extent.

Despite the data analysis efforts, the technique show potential to better understand watershed effects and its distribution without the high computational times at a neighborhood scale. This work will not cover the treatment of time series analysis for all HRUs. Creating the necessary software, preparing, running and calibrating the model are the milestones for this work. **This work required extensive data-science and GIS programming, and the creation of data structures and databases to explore data would not compatible with the time requirement of a Master Thesis**. PySWAT is still under development and a full beta version is expected to be released soon. Also, results are to be better examined until a full work is to be completed under the objectives of this work.

HRUs in SWAT behave as reservoirs. At the end of each timestep, LATQ, GWQ and SURQ are summed algebraically channel routing. The CN zones creation + parameter did not cause significant reductions to most HRU located in subbasins 95 WYLD values. At subbasin 110, where soils where more prone to both storage and release, changes can be observed in the HRU hydrographs.

In both HRUs in Subbasins 95 and 110, observed behavior varies in according to soil type and BMP distribution. In the HRUs on subbasin 95, hydrological interactions show low storage capacity and limited infiltration capacity. As soil moisture is frequently closer to field capacity than the calibrated scenario, infiltration is reached in extreme events. As a consequence, surface runoff reductions reach a maximum value, remaining constant afterwards.

In the BMPs located in subbasin 110, hydrographs show conditions of zero surface runoff for certain reduction values. Although no significant changes are observed, it is shown that for certain HRUs, full surface runoff abatement is possible,despite CN values not being sensitive for the watershed in general.

At the subbasin scale, no significant differences can be observed at subbasin 95 and 110, as shown in the multi-variable HRU plots. Reduced scenarios for all BMPs and cases reach Nash-Sutcliffe Efficiency values close to unity. This is also true for subbasin 110, even though HRUs in subbasin 110 present different dynamics of groundwater recharge and recession. This is due to the magnitudes of values and the temporal and spatial scales. During calibration, CN was not found to be a driving factor for the reach flow, and consequently WYLD. And even though the CN values for urban areas cannot be considered low, its values still do not affect the storage dynamics in the watershed.

At the Subbasin Annual Scale, the most noticeable effects are caused by Swales. Differences are observed for surface runoff and lateral flow. Very small differences are observed in the average soil water, groundwater flow and water yield. The most noticeable SURQ reduction effects were from Swales, followed by Infiltration Basins and Biorretention devices. As for Soil Water, the largest reductions were from Infiltration Basins followed by Swales. Similar results were obtained for Lateral Flow and Groundwater Flow variations. At this scale it is noticeable the spatial distribution of reductions. Although for different parameters the differences are not in the same magnitude, a certain spatial distribution can be seen, and it is particularly corrilated with the soil distribution.

Some reductions can be seen on peak flow on reaches, as shown in Figure 56 to Figure 61. However, these changes on average are very small, often result in correlation coefficients close to unity. Reach water balances receive and route the WYLD components from subbasins. Changes in both HRU and, at smaller scale, at the subbasin scale can be noticed. However, the BMP changes at the subbasin level involve compartment changes that when superimposed with the watershed non-BMP effects, in addition to the boundary conditions obtained from the BMP feasibility section, do not represent significant changes at the reach scale, more specifically, at the stations used for this study.

The effects of creating recharge zones (or, as simplified by this work, BMPs) implementation effects vary according to: (a) how much water is being infiltrated, (b) antecedent HRU moisture (c) groundwater storage antecedent conditions, and (d) Subbasin routing dynamics. Particularly, for all scenarios, SURQ/PREC values reduce in proportion to CN reductions. Furthermore, the effects can be observed in HRUs, although the effect is not propagated to the subbasin scale, being surface runoff not a determinant factor to the quantities of water delivered to the reach.

6 FINAL REMARKS

This work is a response to a challenge: "represent BMPs having with only daily flow data available and assess the effects of a model digitally modified to include lower CN zones, or BMPs". This challenge was overcome by employing: (i) Using average resolution GIS data to generate GIS scenarios as inputs to SWAT (ii) Systematically switch parameters to lower CN conditions and (iii) evaluate the obtained results. This work briefly approached a selected range of results in various scale. The goal achieved in this work is to initially tackle the obtained data, show proof the capability of employing the proposed algorithm and its potential of result delivery, while assessing what are some of the conditions that might be found on the watershed. According to the classification proposed by Harmel et al. (2014) for diffuse pollution modelling result applicability, this study should classified as an exploratory study.

Watershed effect evaluation in urban environments is frequently limited to the difficulties to obtain and input detailed multiple routing parameterization (i.e.: surface flow, open-channel flow and channelized flow) or on the extensive required computational times, which can be an obstacle for environmental motoring and diagnosis. The paradox between timescales, process representation and data requirements on watershed models originated the scientific question that this work seeks to answer: assess benefits and expected outcomes from BMP implementation, and introduce this type of approach on the form of pySWAT

To spatially allocate the BMP devices, the USEPA BMP Sitting tool was used to generate scenarios The PySWATBMPApp applications assist the data input to the generation of SWAT models. The PySWATBMPApp, is programmed using ArcPy. The default terminal has memory consumption limitations(i.e.: all geographical operation information is strored in ArcGIS log. The log, an xml file, consumes computing time and registers are not erased as new log-files are created. ArgGIS crashes when log files are over 2Gb) forcing the user to divide datasets into datasets to which the processing log consumes less than 2Gb per execution and simultaneously not exhausts Hard Drive memory. The PySWATGetIO application assists model calibration, automating the parameter change process using Python code.

SWAT and many other models are legacy code, and in the last decades its developers have focused in developing easy-to-use and scalable model input applications. Another research branch in watershed simulation models is optimizing watershed benefits using evolutionary algorithms. Within the work done thus far, some software has been developed to assist data visualization and parameter calibration. In comparison to what exists, the main advantage on a Python Framework is that operations may be automated at will, and data Input/output becomes compatible with all formats and libraries written in Python, from basic tasks as viewing model results up to more sophisticated tasks as time-series analysis. As the tools manages SWAT simulations and allows for broad data exploration, the potential for integration of data-science platforms is certainly a tool for broader data analysis and model refinement.

The GIS integration with SWAT using ArcSWAT for input writing thresholds the process of data construction, since documentation writing relies in Graphical interfaces that at each run prompts for several inputs via GUI. Certainly, the Graphical interface is the very mechanism that achieves user-friendly inputs and allows to most000 users to experiment with the model. However, implementing an SAOT on SWAT, it would be very difficult (Petersen, G.W., Hamlett, M.J., Baumer, G.M., Miller. D.A.; R.L.,1991) to test various BMP existence and varied performance scenarios using an ArcSWAT input procedure.

The conceptual model in this work comprehended only the largest point source inputs and more relevant do water balance withdrawals. Effects that would be important for resource assessment or creation of regulation are considered negligible for the purposes of this study, which is enlighten the question about how and where BMP effects should be observed and expected and understand possible effects of BMPs on the subbasin and watershed level.

Model calibration is according to literature considered adequate. Multiple calibration attempts were performed. An upstream-downstream procedure was employed to particularize for each incremental subbasin homogeneous behavior. The simulations performed well according to the literature. Time-series efficiency indicators also were acceptable according to the literature, were at the SQ and CX station is equal to **0.53** and **0.60**.

Results at the HRU scale are dependent on HRU parameterization and area percentage rate within its subbasin. At the HRU scale, SWALES provoke changes in SURQ and WYLD in subbasin 95 and more prominently in Subbasin 110, with significant changes to SW, PERC, and Groundwater storage. At the subbasin scale, PERC and SW showed significant increases, which is be positive for groundwater recharge and watershed impermeabilization effect mitigation. SURQ does not decreases significantly at the subbasin. The largest watershed SURQ reductions are observed in the subbasins equipped with swales, followed by Infiltration basins and Biorretention. HRU performance within this study is mainly dependent on (a)BMP area coverage, (b)HRU slopes and drainage lengths, obtained from topographical information, and most importantly (c)On the response to the superimposed effects of the aforementioned features, subjects to the SWAT equations and relationships. Consequently, for the different CN scenarios, HRUs located in urban types of soil were the less affected by the CN changes, being limited to the high soil moisture conditions. A very heterogeneous behavior is shown among subbasins, being the method presented in this work a useful tool to assess watershed behavior in various scales.

Concerning BMPs as an efficient tool at this waterhed, HRU and subbasin hydrographs suggest that the watersheds balances are significantly changed, meaning that more natural conditions can indeed be achieved through BMP implementation. Increases in Soil Water and hence groundwater flow benefits are plant soil water availability, aquifer quality enhancement, peak attenuation at the BMP scale and subbasin scale and the aquatic chemistry benefits of intercepting chemical constituents before reach flow. Another interesting result is that surface runoff can be downgraded to zero, eliminating from channel flow rainfall excess and potentially associated runoff constituents. However, the employment of such devices in such a case would require overflow systems, since soil maximum infiltration is exceeded and runoff is still generated.

The strength of the methods used on this work are: (a) Construction and demonstration of an GIS algorithm to automate scenario generation, which followed by code adaptation could assist BMP hydrological BMP effect assessment, (b) Unifying SWAT relatively affordable data inputs to the GIS scenario generation, which allows to assess mass transfer occurrence and quantification. This work focused in the daily scale, in order to reach balance between computational effort and data availability and (c)Understanding watershed water balances assists to regionally quantify watershed BMP benefits, whether in terms of aquifer storage, flow retention or baseflow increase.

As for its, limitations, (a)BMP CN parameterization relies on the creation of new BMPs. Such features are loaded into SWAT as HRUs with certain parameters (Area, HRU_SLP and SLSOIL and SLSBBSN and) obtained from GIS data (DEM and Area). Once a BMP feature is created, its slope differs from the original HRU slope, as different areas, and hence, average slopes and drainage lengths are likely to be different. SWAT formulation typically does not regard HRU location to be relevant, according to the model WYLD formulation. Different HRU marked areas will result in different average shape and slope. The first consequence of such in this SWAT application, contrary to its original formulation, regards BMP location as important, as different locations will set local values to physical parameters. The second is that, consequently HRU_SLOPE and SLSBBSN are "redefined" as BMP areas are reshaped when constructing a BMP scenario, (b)The Curve Number approach is a simplified interpretation of a very complex phenomena, (c) The application of a daily model to assess results of devices designed for small events could encapsulate significant errors, that in more detailed studies could provide unrealistic results.

Another observation concerning different process timescales is that this work lumps the BMP effects at the daily scale. BMP devices are mainly designed for small to average storms, excepti if they are designed to be direct flood control measures. However, despite undergoing physical processes occurring at more faster scales, employing daily simulations in SWAT to assess watershed benefits and understand its hydrological processes it is recognized as an uncomplicated and rather rich in terms of results manner of proceeding to solve the paradox.

Diffuse pollution monitoring and modelling are important tools to assess how do constituents are transferred between environmental compartments through the hydrological cycle. Contaminant transport and fate watershed models are very important assessment tools to this end. This works aims to contribute with solving challenges in tool availability and watershed effects comprehension of effectively using BMPs on a real scenario.

The next steps in this work will be (i) validating the obtained series, (ii) adapt pySWAT to better work with input/output and structure it to work as a single framework, where all functions may be called and compatible with any model, (iii) use time-series aggregation techniques to best evaluate the results of this model. Despite no water quality processes have been considered, as the original objective of this work, the know-how and a workflow has been established and documented in this work and will be continued. The Barigui watershed has been studied by the Federal University of Parana for several years, and this work is a part of a chain of knowledge about this specific watershed. The watershed is known for having water quality issues. Hence, the continuation of this work is part of a commitment to the more intensely affected communities. (iv) calibrate the model to output water quality variables. (iv) Enhance pySWAT. This work is also the first experiment to the creation of pySWAT, a command-line tool for SWAT, which is expected to greatly enhance input/output analysis.

7 REFERENCES

ABBASPOUR, K. C. Calibration of Hydrologic Models: When is a Model Calibrated? **MODSIM 2005 International Congress on Modelling and Simulation**, p. 2449–2455, 2005.

ABBASPOUR, K. C. SWAT Calibration and Uncertainty Program - A user manual. **EAWAG**, p. 100, 2015.

ABBASPOUR, K. C.; ROUHOLAHNEJAD, E.; VAGHEFI, S.; et al. A continental-scale hydrology and water quality model for Europe: Calibration and uncertainty of a high-resolution large-scale SWAT model. **Journal of Hydrology**, v. 524, p. 733–752, 2015. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.jhydrol.2015.03.027>. .

ABBASPOUR, K.; VAGHEFI, S.; SRINIVASAN, R. A Guideline for Successful Calibration and Uncertainty Analysis for Soil and Water Assessment: A Review of Papers from the 2016 International SWAT Conference. **Water**, v. 10, n. 1, p. 6, 2017. Disponível em: <http://www.mdpi.com/2073-4441/10/1/6>. .

ABDELWAHAB, O. M. M.; RICCI, G. F.; DE GIROLAMO, A. M.; GENTILE, F. Modelling soil erosion in a Mediterranean watershed: Comparison between SWAT and AnnAGNPS models. **Environmental Research**, v. 166, n. December 2017, p. 363–376, 2018. Elsevier Inc. Disponível em: <https://doi.org/10.1016/j.envres.2018.06.029>. .

AGUASPARANA. Dados para download - Bacia do alto Iguaçu. .

AHIABLAME, LAURENT M; A. ENGEL, B.; CHAUBEY, I. Representation and Evaluation of Low Impact Development Practices with L-THIA-LID: An Example for Site Planning. **Environment and Pollution**, v. 1, n. 2, p. 1–13, 2012. Disponível em: <http://www.ccsenet.org/journal/index.php/ep/article/view/14023>. .

AHIABLAME, LAURENT M.; ENGEL, B. A.; CHAUBEY, I. Effectiveness of low impact development practices: Literature review and suggestions for future research. **Water, Air, and Soil Pollution**, v. 223, n. 7, p. 4253–4273, 2012.

AMIN, M. G. M.; VEITH, T. L.; COLLICK, A. S.; KARSTEN, H. D.; BUDA, A. R. Simulating hydrological and nonpoint source pollution processes in a karst watershed: A variable source area hydrology model evaluation. **Agricultural Water Management**, v. 180, p. 212–223, 2017. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.agwat.2016.07.011>. .

ANA. HIDROWEB - Sistema Nacional de informações Hidrológicas. , 2017.

ANDRÉASSIAN, V.; PERRIN, C.; BERTHET, L.; et al. Crash tests for a standardized evaluation of hydrological models. **Hydrology and Earth System Sciences**, v. 13, n. 10, p. 1757–1764, 2009.

ARNOLD, J. G.; FOHRER, N. SWAT2000: Current capabilities and research opportunities in applied watershed modelling. **Hydrological Processes**, v. 19, n. 3, p. 563–572, 2005.

ARNOLD, J. G.; MORIASI, D. N.; GASSMAN, P. W.; et al. Swat: Model Use, Calibration, and Validation. **Asabe**, v. 55, n. 4, p. 1491–1508, 2012.

ASCE. **Guide for Best Management Practices(BMP)selection in Urban developes areas**. I ed. 2015.

BARRETT, M. E.; JR., L. B. I.; JR., J. F. M.; CHARBENEAU, R. J. Characterization of Highway Runoff in Austin, Texas, Area. **Journal of Environmental Engineering**, v. 124, n. 2, p. 131–137, 1998. Disponível em: <http://dx.doi.org/10.1061/(ASCE)0733-9372(1998)124:2(131)>. .

BEKELE, E. G.; NICKLOW, J. W. Multi-objective automatic calibration of SWAT using NSGA-II. **Journal of Hydrology**, v. 341, n. 3–4, p. 165–176, 2007.

BORAH, D. K.; BERA, M. W −s h n −s p m : r a. , v. 47, n. 3, p. 789–804, 2004.

BOSLEY, E. K.; BOSLEY, I. I.; KERN, E. Hydrologic evaluation of low impact development using a continuous, spatially-distributed model. **Civil Engineering**, 2008. Disponível em: <http://scholar.lib.vt.edu/theses/available/etd-07302008-220841/>. .

BRANNAN, K. M.; HEATWOLE, C. D.; DILLAHA, T. A.; MOSTAGHIMI, S.; YAGOW, E. R. Assessing the Accuracy of Using Calibrated HSPF Data Sets in Ungaged Watersheds. , , n. January, 2014.

BRASIL. Resolução CONAMA nº 357 de 17 de março de 2005. **Diário Oficial da União**, , n. 053, p. 58–63, 2005.

BREVNOVA, ELENA, V ELI, R. N. Green-Ampt Infiltration Model Parameter Determination Using SCS Curve Green-Ampt Infiltration Model Parameter Determination Using SCS Curve Number (CN) and Soil Texture Class , and Application to the SCS Runoff Model . n. April, 2015.

BRIGHENTI, T.; BONUMÁ, N.; CHAFFE, P. Calibração hierárquica do modelo swat em uma bacia hidrográfica Catarinense. **Revista Brasileira de Recursos Hídricos**, v. 21, n. 1, p. 53–64, 2016. Disponível em: <http://www.abrh.org.br/SGCv3/index.php?PUB=1&ID=188&SUMARIO=5147>. .

BURTON, G. A.; PITT, R. E. **Stormwater Effects Handbook**. 2001.

CANHOLI, A. P. **Drengem Urbana e controle de enchentes**. I ed. Sao Paulo: Compania das Letras, 2015.

CARPENTER, S.; CARACO, N.; CORRELL, D.; et al. Nonpoint pollution of surface waters with phosphorus and nitrogen. , v. 8, n. December 1997, p. 559–568, 2008. Disponível em:<http://www.esajournals.org/doi/pdf/10.1890/1051-0761(1998)008[0559:NPOSWW]2.0.CO;2>. .

CHANG, M.; MCBROOM, M. W.; SCOTT BEASLEY, R. Roofing as a source of nonpoint water pollution. **Journal of Environmental Management**, v. 73, n. 4, p. 307–315, 2004.

CHEN, L.; WANG, G.; ZHONG, Y.; SHEN, Z. Evaluating the impacts of soil data on hydrological and nonpoint source pollution prediction. **Science of the Total Environment**, v. 563–564, p. 19–28, 2016. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.scitotenv.2016.04.107>. .

CHOW, V. TE; MAIDMENT, DAVID, R.; WAYS, L. M. **Applied Hydrology**. First ed. McGraw-Hiill Book, 1988.

COALIAR. **Resolução nº 04 do comitê das bacias do alto Iguaçu e afluentes do alto ribeira - COALIAR**. 2013.

D. N. MORIASI; J. G. ARNOLD; M. W. VAN LIEW; et al. Model Evaluation Guidelines for Systematic Quantification of Accuracy in Watershed Simulations. **Transactions of the ASABE**, v. 50, n. 3, p. 885–900, 2007. Disponível em: <http://elibrary.asabe.org/abstract.asp??JID=3&AID=23153&CID=t2007&v=50&i=3&T=1>.

DAGGUPATI, P.; KYLE R. DOUGLAS-MANKIN; ALEKSEY Y.SHESHUKOV; L.BARNES, P. TARGETING BMP P LACEMENT USING SWAT EDIMENT YIELD Estimates for Fiels-scale BMPs. **Watershed Management to Improve Water Quality Proceedings, Hyatt Regency Baltimore, Maryland USA**, , n. 711, p. 14–17, 2010.

DAI, C.; QIN, X. S.; TAN, Q.; GUO, H. C. Optimizing best management practices for nutrient pollution control in a lake watershed under uncertainty. **Ecological Indicators**, , n. April, p. 1–13, 2017. Elsevier. Disponível em: <http://dx.doi.org/10.1016/j.ecolind.2017.05.016>. .

DIETZ, M. E. Low impact development practices: A review of current research and recommendations for future directions. **Water, Air, and Soil Pollution**, v. 186, n. 1–4, p. 351–363, 2007.

DUAN, H. F.; LI, F.; YAN, H. Multi-Objective Optimal Design of Detention Tanks in the Urban Stormwater Drainage System: LID Implementation and Analysis. **Water Resources Management**, v. 30, n. 13, p. 4635–4648, 2016.

ELI, R.N; LAMONT, S. J. Curve Numbers and Urban Runoff Modeling - Application Limitations. Low Impact Development 2010. **Anais...** . p.405–418, 2010. Disponível em: <http://link.aip.org/link/ASCECP/v367/i41099/p4/s1&Agg=doi>. .

ELLIOTT, A. H.; TROWSDALE, S. A. A review of models for low impact urban stormwater drainage. **Environmental Modelling and Software**, v. 22, n. 3, p. 394–405, 2007.

EMATER. Mapa de solos do Estado do Paraná. .

EPA., U. S. Preliminary Data Summary of Urban Storm Water Best Management Practices. **Report EPA-821-R-99â€“012**, v. Aug, n. Washington, DC, 1999.

ERTURK, A.; GUREL, M.; BALOCH, M. A.; et al. Applicability of modelling tools in watershed management for controlling diffuse pollution. **Water Science and Technology**, v. 56, n. 1, p. 147–154, 2007.

ESLAMIAN, S. Nonpoint source and water quality modelling. **Handbook of Engineering hydrology: Environmental hydrology and water management**. p.261–299, 2016.

ESRI. ArcGIS Platform - Geostatistical analyst module. , 2015. Redlands, CA.

ESRI. ArcPy - What is ArcPy. Disponível em: <http://desktop.arcgis.com/en/arcmap/10.3/analyze/arcpy/what-is-arcpy-.htm>. .

FANG, Q. Adapting Chinese cities to climate change. **Science**, v. 354, n. October 2016, 2016.

FARZIN, Y. H.; KAPLAN, J. D. Nonpoint source pollution control under incomplete and costly information. **Environmental and Resource Economics**, v. 28, n. 4, p. 489–506, 2004.

FENG, X. Q.; ZHANG, G. X.; JUN XU, Y. Simulation of hydrological processes in the Zhalong wetland within a river basin, Northeast China. **Hydrology and Earth System Sciences**, v. 17, n. 7, p. 2797–2807, 2013.

FERREIRA, D.B.,MÜHLENHOFF, A.P., FERNANDES, C. V. . Modelos de poluição Difusa: Desafios, Estratégias e impacto para a Gestão de Recursos Hídricos. **Revista brasileira de Gestao ambiental**, 2018.

FERREIRA, D. B. .; FERNANDES, C. V. S. . **Nonpoint pollution and best management practices assessment using geographic information systems data – application in the Barigui basin, state of Parana**, 2016. Federal University of Parana.

FERREIRA, D. B.; FERNANDES, C. V. . Avaliação do método médias móveis para separação do escoamento superficial baseado em séries de vazão de alta frequência. . p.1–8, 2013. Bento Goncalves: XX Simposio Brasileiro de Recursos Hidricos. Disponível em: <https://abrh.s3.sa-east-1.amazonaws.com/Sumarios/155/9c2bd0b090886a987c60d5c633cdcecb_8048cf282f43713e7a8723dedf573171.pdf>. .

FERRIER, R. C. Diffuse Pollution - What Is the Nature of the Problem? **Water and Environment Journal**, v. 19, n. 4, p. 74–366, 2005.

FIELD, R.; TARURI, A. N. The Use of Best Management Practices (BMPs) in Urban Watersheds. , , n. September, p. 268, 2005.

FUKUZAKI, N.; YANAKA, T.; URUSHIYAMA, Y. Effects of studded tires on roadside airborne dust pollution in Niigata, Japan. **Atmospheric Environment (1967)**, v. 20, n. 2, p. 377–386, 1986.

GAREN, D. C.; MOORE, D. S. Curve Number Hydrology in Water Quality Modeling : Uses , Abuses , and Future Directions 1. , v. 3224, n. 03127, 2005.

GEZA, M.; MCCRAY, J. E. Effects of soil data resolution on SWAT model stream flow and water quality predictions. **Journal of Environmental Management**, v. 88, n. 3, p. 393–406, 2008.

GIACOMONI, M. H.; JOSEPH, J. Multi-Objective Evolutionary Optimization and Monte Carlo Simulation for Placement of Low Impact Development in the Catchment Scale. **Journal of water resources planning and management**, v. 2, n. 9, p. 1–15, 2017.

GITAU, M. W.; GBUREK, W. J.; L., B. P. Use of the SWAT model to quantify water quality effects of agricultural BMPs at the farmscale- level. **Transactions of the ASABE**, v. 51, n. 2003, p. 1925–1936, 2008.

GONCALVES, M. F. .; FERNANDES, C. V. S. **Variação temporal e espacial da presença dos metais pesados Cd, cr, ni, pb, zn na bacia do rio barigüi e identificação de suas Fontes potenciais**, 2008. Federal University of Parana.

GURUNG, D. P.; GITHINJI, L. J. M.; ANKUMAH, R. O. Assessing the nitrogen and phosphorus loading in the Alabama (USA) River Basin using PLOAD model. **Air, Soil and Water Research**, v. 6, p. 23–36, 2012.

HARMEL, R. D.; SMITH, P. K.; MIGLIACCIO, K. W.; et al. Evaluating, interpreting, and communicating performance of hydrologic/water quality models considering intended use: A review and recommendations. **Environmental Modelling and Software**, v. 57, p. 40–51, 2014. Elsevier Ltd. Disponível em: <http://dx.doi.org/10.1016/j.envsoft.2014.02.013>. .

HER, Y.; JEONG, J.; ARNOLD, J.; et al. A new framework for modeling decentralized low impact developments using Soil and Water Assessment Tool. **Environmental Modelling and Software**, v. 96, p. 305–322, 2017. Elsevier Ltd. Disponível em: <http://dx.doi.org/10.1016/j.envsoft.2017.06.005>. .

HUNT, W. F.; KANNAN, N.; JEONG, J.; GASSMAN, P. W. Stormwater best management practices: Review of current practices and potential incorporation in SWAT. **International Agricultural Engineering Journal**, v. 18, n. 1–2, p. 73–89, 2009. Disponível em: <https://beta.engineeringvillage.com/share/document.url?mid=cpx_535b58126475f39a0M6a2c2061377553&database=cpx>. .

IPPUC. IPPUC - Base de dados geoespaciais. .

JEONG, J.; KANNAN, N.; ARNOLD, J.; et al. Development and Integration of Sub-hourly Rainfall-Runoff Modeling Capability Within a Watershed Model. **Water Resources Management**, v. 24, n. 15, p. 4505–4527, 2010.

JEONG, J.; KANNAN, N.; ARNOLD, J. G.; et al. Modeling Sedimentation-Filtration Basins for Urban Watersheds Using Soil and Water Assessment Tool. , v. 139, n. June, p. 838–848, 2013.

KAINI, P.; ARTITA, K.; NICKLOW, J. W. Evaluating Optimal Detention Pond Locations at a Watershed Scale. **World Environmental and Water Resources Congress 2007**, , n. 618, p. 1–8, 2007. Disponível em: <http://ascelibrary.org/doi/abs/10.1061/40927(243)170>. .

KELLAGHER, R.; MARTIN, P.; JEFFERIES, C.; et al. **The SUDS manual**. 2015.

KLEMEŠ, V. Operational testing of hydrological simulation models. **Hydrological Sciences Journal**, v. 31, n. 1, p. 13–24, 1986.

KOO, B. K.; O'CONNELL, P. E. An integrated modelling and multicriteria analysis approach to managing nitrate diffuse pollution: 2. A case study for a chalk catchment in England. **Science of the Total Environment**, v. 358, n. 1–3, p. 1–20, 2006.

KOZAK, C. ; FERNANDES, C. V. S. **Water quality aassessment and its effects on diffuse pollution considering a new wter quality and quantity approach**, 2016. Universidade Federal do Parana.

LANE, S. N.; BROOKES, C. J.; HEATHWAITE, A. L.; REANEY, S. Surveillant science: Challenges for the management of rural environments emerging from the new generation diffuse pollution models. **Journal of Agricultural Economics**, v. 57, n. 2, p. 239–257, 2006.

LERNER, D. N. Identifying and quantifying urban recharge: A review. **Hydrogeology Journal**, v. 10, n. 1, p. 143–152, 2002.

LI, S.; ZHUANG, Y.; ZHANG, L.; DU, Y.; LIU, H. Worldwide performance and trends in nonpoint source pollution modeling research from 1994 to 2013: A review based on bibliometrics. **Journal of Soil and Water Conservation**, v. 69, n. 4, p. 121A-126A, 2014. Disponível em: <http://www.jswconline.org/cgi/doi/10.2489/jswc.69.4.121A>. .

LIN, C.; MA, R.; ZHU, Q.; LI, J. Using hyper-spectral indices to detect soil phosphorus concentration for various land use patterns. **Environmental monitoring and assessment**, v. 187, n. 1, p. 4130, 2015.

LIN, C.; WU, Z.; MA, R.; SU, Z. Detection of sensitive soil properties related to non-point phosphorus pollution by integrated models of SEDD and PLOAD. **Ecological Indicators**, v. 60, p. 483–494, 2016. Elsevier Ltd.

LIU, Y.; AHIABLAME, L. M.; BRALTS, V. F.; ENGEL, B. A. Enhancing a rainfall-runoff model to assess the impacts of BMPs and LID practices on storm runoff. **Journal of Environmental Management**, v. 147, p. 12–23, 2015. Elsevier Ltd. Disponível em: <http://dx.doi.org/10.1016/j.jenvman.2014.09.005>. .

LIU, Y.; ENGEL, B. A.; FLANAGAN, D. C.; et al. A review on effectiveness of best management practices in improving hydrology and water quality: Needs and opportunities. **Science of the Total Environment**, v. 601–602, p. 580–593, 2017. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.scitotenv.2017.05.212>. .

MAIDMENT, DAVID, R. **Handbook of hydrology**. First Edit ed. McGraw-Hiill Book, 1993.

MALAGÒ, A.; EFSTATHIOU, D.; BOURAOUI, F.; et al. Regional scale hydrologic modeling of a karst-dominant geomorphology: The case study of the Island of Crete. **Journal of Hydrology**, v. 540, p. 64–81, 2016.

MAO, X.; JIA, H.; YU, S. L. Assessing the ecological benefits of aggregate LID-BMPs through modelling. **Ecological Modelling**, v. 353, n. April 2015, p. 139–149, 2017. Elsevier B.V.

MARINGANTI, C.; CHAUBEY, I.; POPP, J. Development of a multiobjective optimization tool for the selection and placement of best management practices for nonpoint source pollution control. **Water Resources Research**, v. 45, n. 6, 2009.

MEIN, R.G., LARSON, C. L. Rainfall intensity is initially less than the infiltration capacity of the soil. In this paper a. **Water Resources Research**, v. 9, n. 2, p. 384–394, 1973.

MERRILL, L. S.; CRAWFORD, S. K.; HALL, T. Manual of best management practices (bmps) for agriculture in new hampshire. , v. 1, n. June, 2012.

MIGNOT, E.; LI, X.; DEWALS, B. Experimental modelling of urban flooding: A review. **Journal of Hydrology**, v. 568, n. November 2018, p. 334–342, 2019.

MILLEDGE, D. G.; LANE, S. N.; HEATHWAITE, A. L.; REANEY, S. M. A Monte Carlo approach to the inverse problem of diffuse pollution risk in agricultural catchments. **Science of the Total Environment**, v. 433, p. 434–449, 2012. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.scitotenv.2012.06.047>. .

MOLINA-NAVARRO, E.; ANDERSEN, H. E.; NIELSEN, A.; THODSEN, H.; TROLLE, D. The impact of the objective function in multi-site and multi-variable calibration of the SWAT model. **Environmental Modelling and Software**, v. 93, p. 255–267, 2017. Elsevier Ltd. Disponível em: <http://dx.doi.org/10.1016/j.envsoft.2017.03.018>. .

MOTSINGER, J.; KALITA, P.; BHATTARAI, R. Analysis of Best Management Practices Implementation on Water Quality Using the Soil and Water Assessment Tool. **Water**, v. 8, n. 4, p. 145, 2016. Disponível em: <http://www.mdpi.com/2073-4441/8/4/145>. .

MUHLENHOFF, A. P.; FERNANDES, C. V. S. . **AVALIAÇÃO DE POLUIÇÃO DIFUSA EM BACIAS URBANAS**, 2012. Federal University of Parana.

MUNAFÒ, M.; CECCHI, G.; BAIOCCO, F.; MANCINI, L. River pollution from non-point sources: A new simplified method of assessment. **Journal of Environmental Management**, v. 77, n. 2, p. 93–98, 2005.

NASH, J. E.; SUTCLIFFE, J. V. River Flow Forecasting Through Conceptual Models Part I-a Discussion of Principles*. **Journal of Hydrology**, v. 10, p. 282–290, 1970.

NEITSCH, S. .; ARNOLD, J. .; KINIRY, J. .; WILLIAMS, J. . Soil & Water Assessment Tool Theoretical Documentation Version 2009. **Texas Water Resources Institute**, p. 1–647, 2011.

NOVOTNY, V. Diffuse pollution from agriculture - A worldwide outlook. **Water Science and Technology**, v. 39, n. 3, p. 1–13, 1999.

NOVOTNY, V. **Diffuse pollution and Watershed management**. I ed. United States, 2003.

NRCS. Urban Hydrology for Small Watersheds TR-55. **USDA Natural Resource Conservation Service Conservation Engeneering Division Technical Release 55**, p. 164, 1986.

DE OLIVEIRA, L. M.; MAILLARD, P.; DE ANDRADE PINTO, E. J. Application of a land cover pollution index to model non-point pollution sources in a Brazilian watershed. **Catena**, v. 150, p. 124–132, 2017. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.catena.2016.11.015>. .

OMERNIK, J. M. The influence on land use on stream nutrient levels. EPA-600/3-76-104. U.S. Environ. Prot. Agency, Corvallis, OR. , 1976.

ÖZCAN, Z.; BAŞKAN, O.; DÜZGÜN, H. Ş.; KENTEL, E.; ALP, E. A pollution fate and transport model application in a semi-arid region: Is some number better than no number? **Science of the Total Environment**, v. 595, p. 425–440, 2017.

ÖZCAN, Z.; KENTEL, E.; ALP, E. Evaluation of the best management practices in a semi-arid region with high agricultural activity. **Agricultural Water Management**, v. 194, p. 160–171, 2017.

PANAGOPOULOS, Y.; MAKROPOULOS, C.; BALTAS, E.; MIMIKOU, M. SWAT parameterization for the identification of critical diffuse pollution source areas under data limitations. **Ecological Modelling**, v. 222, n. 19, p. 3500–3512, 2011. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.ecolmodel.2011.08.008>. .

PATIL, A.; RAMSANKARAN, R. A. A. J. Improving streamflow simulations and forecasting performance of SWAT model by assimilating remotely sensed soil moisture observations. **Journal of Hydrology**, v. 555, p. 683–696, 2017. Elsevier B.V. Disponível em: <https://doi.org/10.1016/j.jhydrol.2017.10.058>. .

PEREZ-VALDIVIA, C.; CADE-MENUN, B.; MCMARTIN, D. W. Hydrological modeling of the pipestone creek watershed using the Soil Water Assessment Tool (SWAT): Assessing impacts of wetland drainage on hydrology. **Journal of Hydrology: Regional Studies**, v. 14, n. September, p. 109–129, 2017. Elsevier. Disponível em: <https://doi.org/10.1016/j.ejrh.2017.10.004>. .

PETERSEN, G.W., HAMLETT, M.J., BAUMER, G.M., MILLER. D.A., D.; R.L., R. J. M. **Evaluation of agricultural nonpoint pollution potential in Pennsylvania using a geographic information system**. Harrisburg, PA, 1991.

PORTLAND, C. OF. Stormwater Management Manual. , p. 502, 2016.

RAHMAN, M. M.; THOMPSON, J. R.; FLOWER, R. J. An enhanced SWAT wetland module to quantify hydraulic interactions between riparian depressional wetlands, rivers and aquifers. **Environmental Modelling and Software**, v. 84, p. 263–289, 2016. Elsevier Ltd. Disponível em: <http://dx.doi.org/10.1016/j.envsoft.2016.07.003>.

ROMAGNOLI, M.; PORTAPILA, M.; RIGALLI, A.; et al. Assessment of the SWAT model to simulate a watershed with limited available data in the Pampas region, Argentina. **Science of the Total Environment**, v. 596–597, p. 437–450, 2017. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.scitotenv.2017.01.041>.

SAMPLE, D. J.; HEANEY, J. P.; WRIGHT, L. T.; KOUSTAS, R. Geographic information systems, decision support systems and urban storm-water management. **Journal of Water Resources Planning and Management**, v. 127, n. June, p. 155–161, 2001.

SANEPAR. Santa Quiteria WWTP fact sheets. , 2017. Curitiba.

SEBTI, A.; CARVALLO ACEVES, M.; BENNIS, S.; FUAMBA, M. Improving Nonlinear Optimization Algorithms for BMP Implementation in a Combined Sewer System. **Journal of Water Resources Planning and Management**, v. 142, n. 9, p. 04016030, 2016.

SEMA, A. **Qualidade das Águas da bacia hidrográfica do Alto Iguaçu: 1987-2010**. 2014.

SHANNAK, S. The effects of green infrastructure on exceedance of critical shear stress in Blunn Creek watershed. **Applied Water Science**, v. 7, n. 6, p. 2975–2986, 2017. Springer Berlin Heidelberg. Disponível em: <http://link.springer.com/10.1007/s13201-017-0606-5>. .

SHAVER, E.; HORNER, R.; SKUPIEN, J.; MAY, C.; RIDLEY, G. Fundamentals of urban runoff management: technical and institutional issues. **Change**, p. 327, 1994. Disponível em: <http://agris.fao.org/agris-search/search/display.do?f=1995/US/US95046.xml;US9536404>. .

SHEKHAR, S.; XIONG, H. Soil and Water Assessment Tool “SWAT”. **Encyclopedia of GIS**, p. 1068–1068, 2008. Disponível em: <http://link.springer.com/10.1007/978-0-387-35973-1_1231>. .

SHEN, Z.; CHEN, L.; XU, L. A Topography Analysis Incorporated Optimization Method for the Selection and Placement of Best Management Practices. **PLoS ONE**, v. 8, n. 1, p. 1–12, 2013.

SHEN, Z.; HONG, Q.; CHU, Z.; GONG, Y. A framework for priority non-point source area identification and load estimation integrated with APPI and PLOAD model in Fujiang Watershed, China. **Agricultural Water Management**, v. 98, n. 6, p. 977–989, 2011. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.agwat.2011.01.006>. .

SHEN, Z. Y.; CHEN, L.; LIAO, Q.; LIU, R. M.; HUANG, Q. A comprehensive study of the effect of GIS data on hydrology and non-point source pollution modeling. **Agricultural Water Management**, v. 118, p. 93–102, 2013. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.agwat.2012.12.005>. .

SHESHUKOV, A. Y.; DOUGLAS-MANKIN, K. R.; SINNATHAMBY, S.; DAGGUPATI, P. Pasture BMP effectiveness using an HRU-based subarea approach in SWAT. **Journal of Environmental Management**, v. 166, p. 276–284, 2016. Elsevier Ltd. Disponível em: <http://dx.doi.org/10.1016/j.jenvman.2015.10.023>. .

SHI, Q.; DENG, X.; WU, F.; ZHAN, J.; XU, L. Best management practices for agricultural non-point source pollution control using PLOAD in Wuliangsuhai watershed. **Journal of Food Agriculture & Environment**, v. 10, n. 2, p. 1389–1393, 2012.

SHOEMAKER, L.; DAI, T.; KOENIG, J.; HANTUSH, M. TMDL Model Evaluation and Research Needs. , , n. November, p. 1–403, 2005.

SIMEPAR. Dados hidrológicos da Estação Curitiba. , 2017. Curitiba.

SONG, J. Y.; CHUNG, E. A Multi-Criteria Decision Analysis System for Prioritizing Sites and Types of Low Impact Development Practices: Case of Korea. **Water**, v. 9, n. 4, p. 291, 2017.

SRIVASTAVA, PUNEET, HAMLETT, JAMES M., ROBILLARD, P. D. Watershed optimization of agricultural best management Practices: continuous simulation versus design storms. **Journal of the american water resources association**, v. 39, n. 5, p. 83–93, 2003.

SWAT USER GROUP. Best of SWAT-CUP questions. , p. 1–196, 2013.

TETRA TECH INC. BMP Siting Tool: Step-by-Step Guide. , v. 1, n. January, 2013.

TOBISZEWSKI, M.; POLKOWSKA, Z.; KONIECZKA, P.; NAMIEŚNIK, J. Roofing materials as pollution emitters -concentration changes during runoff. **Polish Journal of Environmental Studies**, v. 19, n. 5, p. 1019–1028, 2010.

TUCCI, C. E. M. Urban Waters. **Advanced Studies**, v. 22, n. 63, p. 97–112, 2010. Disponível em: <http://www.ecy.wa.gov/urbanwaters/spokaneriver.html>. .

U.S. EPA. Results of the nationwide urban runoff program, vol. I. Final Report. **United States Environmental Protection Agency**, 1983. Disponível em: <http://www3.epa.gov/npdes/pubs/sw_nurp_vol_1_finalreport.pdf>. .

UNITED STATES FEDERAL LAW. Federal Water Pollution Control Act Amendments of 1961. **Public health reports (Washington, D.C. : 1896)**, v. 77, p. 107–13, 1962. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/13880338%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1914595>. .

USEPA. National Pollution Discharge Elimination System. Disponível em: <https://www.epa.gov/npdes>. Acesso em: 13/4/2017.

VIJI, R.; PRASANNA, P. R.; ILANGOVAN, R. Modified SCS-CN and Green-Ampt Methods in Surface Runoff Modelling for the Kundahpallam Watershed, Nilgiris, Western Ghats, India. **Aquatic Procedia**, v. 4, n. Icwrcoe, p. 677–684, 2015. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.aqpro.2015.02.087>. .

WAIDLER, D.; WHITE, M.; STEGLICH, E.; et al. Conservation Practice Modeling Guide for SWAT and APEX. **Journal of Hydrology**, v. 50, n. 1–2, p. 30–49, 2011. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0022169410006554%0Ahttp://linkinghub.elsevier.com/retrieve/pii/S1474706512001076%0Ahttp://linkinghub.elsevier.com/retrieve/pii/S002216941300098X%0Ahttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A>. .

WANG, R.; KALIN, L. Modelling effects of land use/cover changes under limited data. **Ecohydrology**, v. 4, n. 2, p. 265–276, 2011.

WANG, X.; SHANG, S.; QU, Z.; et al. Simulated wetland conservation-restoration effects on water quantity and quality at watershed scale. **Journal of Environmental Management**, v. 91, n. 7, p. 1511–1525, 2010. Elsevier Ltd. Disponível em: <http://dx.doi.org/10.1016/j.jenvman.2010.02.023>. .

WANG, X.; YANG, W.; MELESSE, A. M. Using hydrologic equivalent wetland concept within SWAT to estimate streamflow in watersheds with numerous wetlands. , v. 51, n. 1, p. 55–72, 2008.

WARDROPPER, C. B.; GILLON, S.; RISSMAN, A. R. Uncertain monitoring and modeling in a watershed nonpoint pollution program. **Land Use Policy**, v. 67, n. July, p. 690–701, 2017. Elsevier. Disponível em: <http://dx.doi.org/10.1016/j.landusepol.2017.07.016>. .

WILLIS, W. B.; EICHINGER, W. E.; PRUEGER, J. H.; et al. Lidar method to estimate emission rates from extended sources. **Journal of Atmospheric and Oceanic Technology**, v. 34, n. 2, p. 335–345, 2017.

WINCHELL, M.; SRINIVASAN, R.; DI LUZIO, M.; ARNOLD, J. ArcSWAT Manual. **Texas Agrilife Research, United States Department of Agriculture, Agricultural Reseach Service**, 2013.

WOODS-BALLARD, B.; KELLAGHER, R.; MARTIN, P.; et al. **The SUDS manual**. 2007.

XU, F.; DONG, G.; WANG, Q.; et al. Impacts of DEM uncertainties on critical source areas identification for non-point source pollution control based on SWAT model. **Journal of Hydrology**, v. 540, p. 355–367, 2016. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.jhydrol.2016.06.019>. .

YANG, Y. S.; WANG, L. A review of modelling tools for implementation of the EU water framework directive in handling diffuse water pollution. **Water Resources Management**, v. 24, n. 9, p. 1819–1843, 2010.

YANHUA, Z.; THUMINH, N.; BEIBEI, N.; EI, S.; SONG, H. Research Trends in Non Point Source during 1975-2010. **Physics Procedia**, v. 33, p. 138–143, 2012. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S1875389212013545>. .

YU, Z. SWAT Output Viewer. , 2015. Disponível em: <https://swatviewer.com/>. .

ZHANG, K.; CHUI, T. F. M. A comprehensive review of spatial allocation of LID-BMP-GI practices: Strategies and optimization tools. **Science of the Total Environment**, v. 621, p. 915–929, 2018. Elsevier B.V. Disponível em: <https://doi.org/10.1016/j.scitotenv.2017.11.281>. .

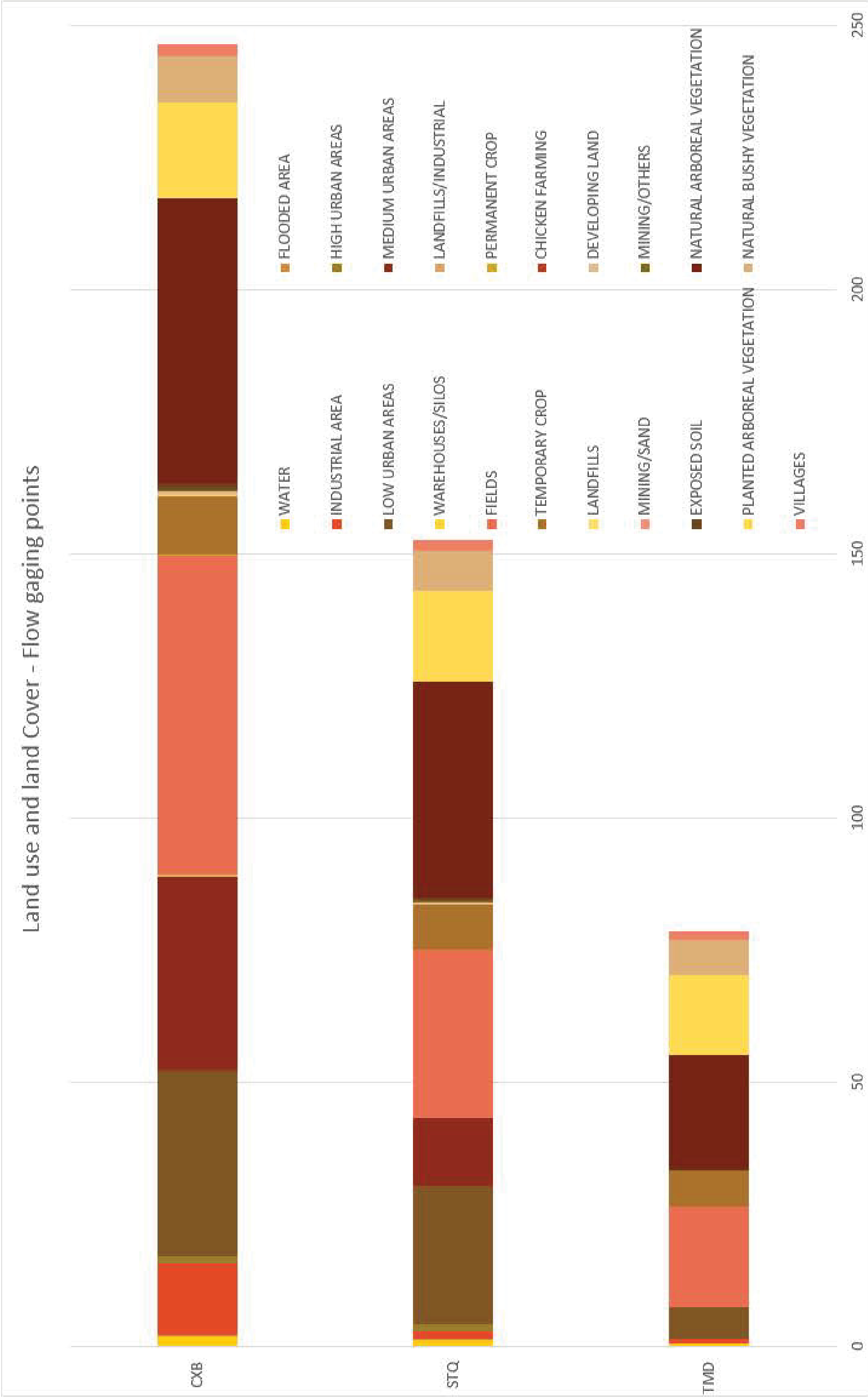
ZOBRIST, J.; MÜLLER, S. R.; AMMANN, A.; et al. Quality of roof runoff for groundwater infiltration. **Water Research**, v. 34, n. 5, p. 1455–1462, 2000.

8 ANNEX

ANNEX 1 – LULC distribution in the Barigui river Basin

OID	LULC CLASS	TMD - km²	STQ - km²	CXB - km²
1	WATER	0.538	1.301	1.918
2	FLOODED AREA	0.114	0.114	0.121
3	INDUSTRIAL AREA	0.816	1.512	13.529
4	HIGH URBAN AREAS	0.000	1.341	1.496
5	LOW URBAN AREAS	6.058	26.151	35.083
6	MEDIUM URBAN AREAS	0.000	12.840	36.694
7	WAREHOUSES/SILOS	0.000	0.000	0.324
8	LANDFILLS/INDUSTRIAL	0.000	0.000	0.038
9	FIELDS	18.896	31.922	60.526
10	PERMANENT CROP	0.016	0.016	0.050
11	TEMPORARY CROP	6.593	8.432	11.172
12	CHICKEN FARMING	0.009	0.040	0.040
13	LANDFILLS	0.011	0.026	0.072
14	DEVELOPING LAND	0.000	0.331	0.751
15	MINING/SAND	0.000	0.000	0.002
16	MINING/OTHERS	0.258	0.258	0.258
17	EXPOSED SOIL	0.114	0.399	1.156
18	NATURAL ARBOREAL VEGETATION	21.765	41.117	54.045
19	PLANTED ARBOREAL VEGETATION	15.073	17.151	18.187
20	NATURAL BUSHY VEGETATION	6.567	7.624	8.691
21	VILLAGES	1.694	2.083	2.377
	Total (km²)	**78.522**	**152.658**	**246.531**

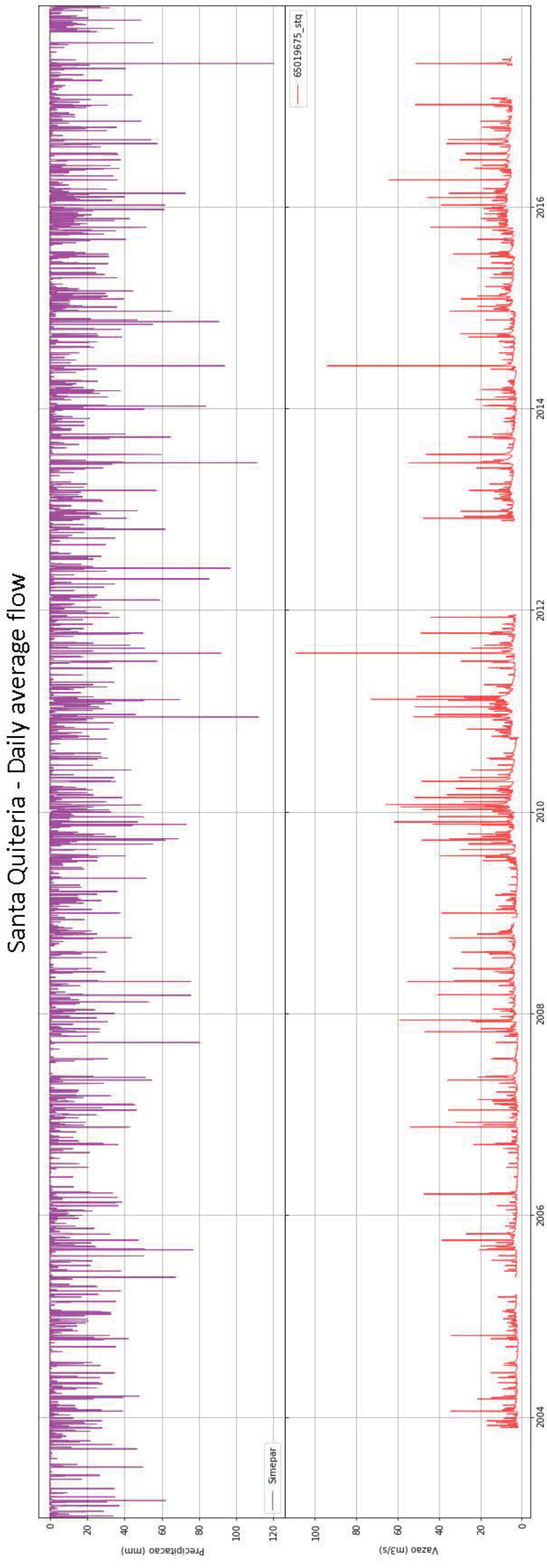
ANNEX 2 – LULC area distribution in the Barigui river Basin(graphic)



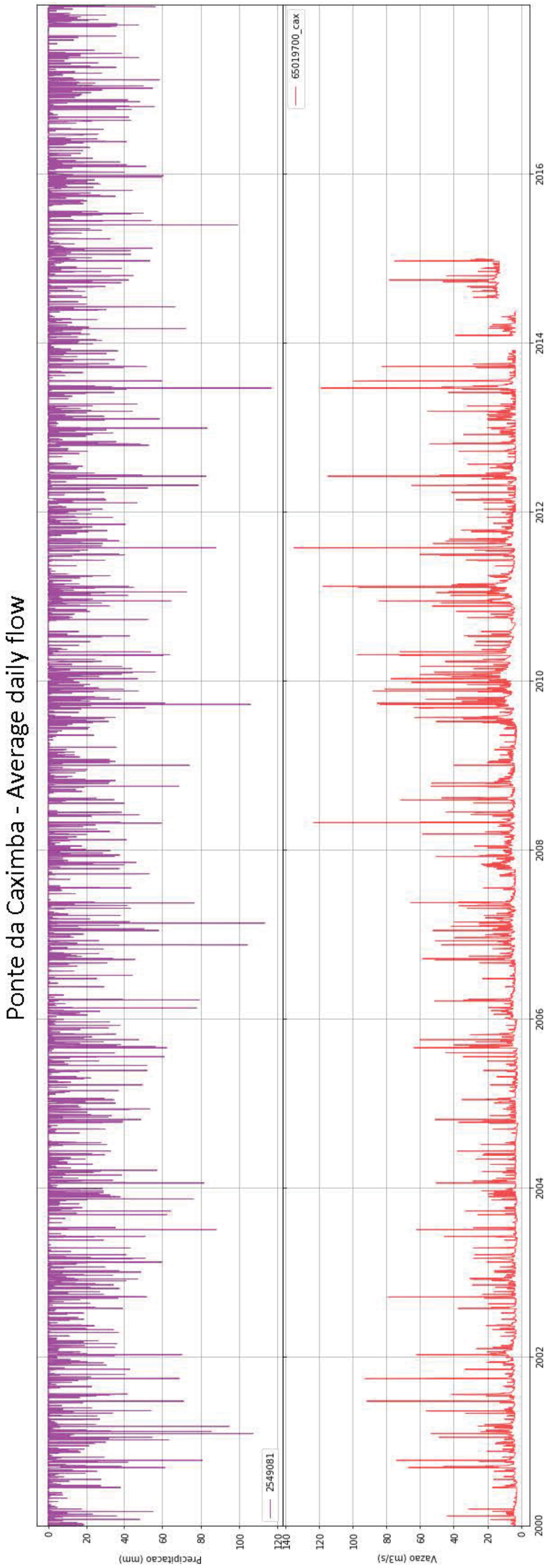
ANNEX 3 – Reference table for LULC reclassification

OID	LULC CLASS	Reclassified LULC
1	WATER	WATR
2	FLOODED AREA	WATR
3	INDUSTRIAL AREA	INDU
4	HIGH URBAN AREAS	RESD
5	LOW URBAN AREAS	RESD
6	MEDIUM URBAN AREAS	INDU
7	WAREHOUSES/SILOS	INDU
8	LANDFILLS/INDUSTRIAL	INDU
9	FIELDS	AGRL
10	PERMANENT CROP	AGRL
11	TEMPORARY CROP	AGRL
12	CHICKEN FARMING	AGRL
13	LANDFILLS	INDU
14	DEVELOPING LAND	RESD
15	MINING/SAND	INDU
16	MINING/OTHERS	INDU
17	EXPOSED SOIL	RESD
18	NATURAL ARBOREAL VEGETATION	FRSE
19	PLANTED ARBOREAL VEGETATION	FRSE
20	NATURAL BUSHY VEGETATION	FRSE
21	VILLAGES	FRSE

ANNEX 5 – SQ station daily flow records with rainfall from Simepar



ANNEX 6 – CX station daily flow records with rainfall from 02549081



ANNEX 7 – PySwatBmpApp-Swales Code

```
# -*- coding: utf-8 -*-
#Dependancies
import os
import archook
archook.get_arcpy()
import arcpy
import arcgisscripting, sys, traceback
import math
import os
import numpy as np
arcpy.env.overwriteOutput = True
arcpy.CheckOutExtension("Spatial")

################   PySwatBmpApp-Swales   ##################################
"""The PySwatBmpApp-Swales Application is an ArcPy application for
(i)mirroring(mirror function) and (ii)resizing (matchsize fuction)
of curbside swales.

AUTHOR: DAVID BISPO FERREIRA(davidbispo@hotmail.com)
SUPPORT: FEDERAL UNIVERSITY OF PARANA(UFPR)/WATER RESOURCES AND ENVIRONMENTAL
ENGINEERING POST-GRADUATE PROGRAM(PPGERHA)/
INFRASTRUCTURE AND TRANSPORTATION TECHNOLOGY INSTITUTE(ITTI)

The mirror function requires as INPUTS
(i) a swale feature layer fileaddress As String
(ii) a street layout feature layer fileaddress As String
(iii) an output fileaddress As String
The mirror function (a)fetches the original polygon area (b)uses successive
buffer+ERASE operations
to reconstruct the original BMP area on the other side of the street layout
polygon(since swales must be
on a curb according to the theoretical formulation of the program).

*The mirror function returns as OUTPUTS:
A mirrored swale feature layer As ArcGIS Feature gdb feature layer or shapefile

The matchsize function requires:
(i) a swale feature layer fileaddress As String
(ii) a street layout feature layer fileaddress As String
(iii) an output fileaddress As String
The matchsize function (a)fetches the target polygon area, using the
formulation proposed by Ferreira(2019) and
(b)uses successive buffer+ERASE operations to reconstruct the target BMP area
on the other side of the street layout polygon(since swales must be
on a curb according to the theoretical formulation of the program).
"""
#Overwrites previous homonimous files
arcpy.overwriteOutput = True

def mirror(swales_feature, streets, output, env, tolerance = 2):

    """ROUTINE FOR MIRRORING SWALES BASED ON A GIVEN BARRIER LAYER"""
#####START####
##Input data##

    arcpy.env.workspace=env
    boundaries = streets
    polygons = swales_feature
    output_feature_class = output
    ok_diff = tolerance
```

```
    gp = arcgisscripting.create()

#creates the final output feature class ->'swales_mirrored'
    arcpy.CreateFeatureclass_management(out_path=arcpy.env.workspace,
        out_name=output_feature_class, geometry_type='POLYGON',
        spatial_reference=arcpy.Describe(polygons).spatialReference)

#creates a street layout feature layer ->'blyr'
    arcpy.MakeFeatureLayer_management(in_features=boundaries, out_layer='blyr')
    counter = 1
#creates cursor in the bmp polygon
    with arcpy.da.SearchCursor(polygons,['OID@','SHAPE@']) as cursor:
        for row in cursor:
            gp.AddOutputsToMap = False

            bufferstart = 4.0
#sql query for creation of layer with the current row in cursor
            sql = """{0} = {1}""".format(
                arcpy.AddFieldDelimiters(polygons,
                arcpy.Describe(polygons).OIDFieldName),row[0])

#creates single bmp layer with current sql selection -> 'polygonlyr'
            arcpy.MakeFeatureLayer_management(in_features=polygons,
                  out_layer='polygonlyr', where_clause=sql)
#gets original bmp area before crossing street
            original_bmp_area = [i[0] for i in arcpy.da.SearchCursor(
                'polygonlyr','SHAPE@AREA')][0]
            print("Original BMP Area = {0}".format(original_bmp_area))

            print("Buffer to enlarge = {0}".format(bufferstart))
            arcpy.Buffer_analysis(in_features='polygonlyr',
                  out_feature_class=r'in_memory\buffered_swale',
                  buffer_distance_or_field="{0} Meters".format(bufferstart))

            arcpy.Erase_analysis(in_features=r'in_memory\buffered_swale',
                 erase_features='blyr',
                 out_feature_class=r'in_memory\clipswale_base')
#gets the buffered feature area
            buffered_bmp_area = [i[0] for i in arcpy.da.SearchCursor(
            r'in_memory\clipswale_base','SHAPE@AREA')][0]
            print("Buffered BMP Area = {0}".format(buffered_bmp_area))

#loop initial settings
            target_area_bmp = original_bmp_area
            polygon_this_area = buffered_bmp_area

            bufferlower = 0.
            bufferupper = -6.
            print("Starting to reconstruct Swale")
            while abs(polygon_this_area-target_area_bmp)>ok_diff:
#calculates the itertation buffer values

                bufferstart = (bufferlower + bufferupper)/2

                print("Difference for Iteration = {0} m2".format(
                    abs(polygon_this_area-target_area_bmp)))
                print("max= {0}, min = {1}. This Buffer = {2}".format(
                    bufferupper, bufferlower, bufferstart))

#Print iteration results
```

```
                    arcpy.Buffer_analysis(in_features=r'in_memory\clipswale_base',
                         out_feature_class=r'in_memory\swale_to_erase',
                         buffer_distance_or_field="{0} Meters".format(bufferstart))

                    if bufferstart>= 0:
                        pass
                    arcpy.Erase_analysis(in_features=r'in_memory\swale_to_erase',
                    erase_features='blyr',
                    out_feature_class=r'in_memory\adjusted_swale')

                    try:
                        polygon_this_area = [l[0] for l in arcpy.da.SearchCursor(
                        r'in_memory\adjusted_swale','SHAPE@AREA')][0]

                        print("buffer successful")

                        print("Iteration BMP area = {0}".format(polygon_this_area))
####Bissection method variable change
                        if polygon_this_area > target_area_bmp:
                            bufferlower = bufferstart

                        elif polygon_this_area < target_area_bmp:
                            bufferupper = bufferstart

                        if polygon_this_area - bufferupper == 0.05:
                            bufferupper -= 1
                            print('too close to the limit')
                    except:
                        print('Buffer failed')
                        if abs(bufferstart) <= 1.0:
                            bufferupper += 0.04
                        if abs(bufferstart) <= 0.5:
                            bufferupper += 0.025
                            print('Buffer reduced 0.04')
                        else:
                            print('Buffer reduced 0.25')
                            bufferupper += 0.20
                        if bufferstart>= 0:
                            pass
#After successfully finding the buffer value, evolves loop for feature counter and appends it tc
                gp.AddOutputsToMap = True
                arcpy.Append_management(inputs=r'in_memory\adjusted_swale',
                    target=output_feature_class, schema_type='NO_TEST')
                print('*************Feature {0} done************!'.format(counter))
                counter += 1
#####END####

def matchsize(swales_feature, streets, output, env, tolerance = 2):
    """ROUTINE FOR INCREASING SWALE SIZE"""
#####START#####
#Input data ##
    arcpy.env.workspace=env
    Swales = swales_feature
    boundaries = streets
    polygons = Swales
    output_feature_class = output
    ok_diff = tolerance
    gp = arcgisscripting.create()
```

```
#creates the final output feature class ->'swales_enlarged'
    arcpy.CreateFeatureclass_management(out_path=arcpy.env.workspace,
        out_name=output_feature_class, geometry_type='POLYGON',
        spatial_reference=arcpy.Describe(polygons).spatialReference)
#creates a street layout feature layer ->'blyr'
    arcpy.MakeFeatureLayer_management(in_features=boundaries, out_layer='blyr')
#creates cursor in the bmp polygon
    with arcpy.da.SearchCursor(polygons,['OID@','SHAPE@']) as cursor:
        for row in cursor:
            gp.AddOutputsToMap = False
            bufferlower = 0.
            bufferupper = 600.
#sql query for creation of layer with the current row in cursor
            sql = """{0} = {1}""".format(
                arcpy.AddFieldDelimiters(
                polygons,arcpy.Describe(polygons).OIDFieldName),row[0])

            print(sql)
#creates single bmp layer with current sql selection -> 'polygonlyr'
            arcpy.MakeFeatureLayer_management(in_features=polygons,
              out_layer='polygonlyr',where_clause=sql)
#gets original bmp area before enlarging swale feature
            swale_initial_area = [i[0] for i in arcpy.da.SearchCursor(
                'polygonlyr','SHAPE@AREA')][0]
            print("Starting BMP area = {0}".format(swale_initial_area))
#gets target area for enlarging Swale feature
            wq_area_for_bmp = 0.15*swale_initial_area/0.004

            target_area_bmp = wq_area_for_bmp
            print("Target BMP area/WQ Area = {0}".format(target_area_bmp))
#loop initial settings
            polygon_area = [k[0] for k in arcpy.da.SearchCursor('polygonlyr',
                'SHAPE@AREA')][0]
            bufferstart = (bufferlower + bufferupper)/2
            feature_counter = 0
            counter = 0

            while abs(polygon_area-target_area_bmp)>ok_diff:

                if counter ==0:
#calculates the itertation buffer values
                    bufferstart = (bufferlower + bufferupper)/2

                    print("Difference = {0}".format(
                        abs(polygon_area-target_area_bmp)))

                    print("Buffer for this iteration = {0}".format(
                            bufferstart))

                    arcpy.Buffer_analysis(in_features='polygonlyr',
                          out_feature_class=r'in_memory\polygon',
                          buffer_distance_or_field="{0} Meters".format(
                          bufferstart))

                    arcpy.Erase_analysis(in_features=r'in_memory\polygon',
                         erase_features='blyr',
                         out_feature_class=r'in_memory\clipbuffer')
#gets the buffered feature area
                    polygon_this_area = [l[0] for l in arcpy.da.SearchCursor(
                        'in_memory\clipbuffer','SHAPE@AREA')][0]
```

```
#Print iteration results
                    print("Iteration BMP area = {0}".format(polygon_this_area))

####Bissection method variable change
                    if polygon_this_area < target_area_bmp:
                        bufferlower = bufferstart

                    elif polygon_this_area > target_area_bmp:
                        bufferupper = bufferstart

                    polygon_area = polygon_this_area

                    counter = counter+1

                else:

                    bufferstart = (bufferlower + bufferupper)/2

                    print("Difference = {0}".format(abs(
                        polygon_area-target_area_bmp)))
                    print("Buffer for this iteration = {0}".format(
                        bufferstart))

                    arcpy.Buffer_analysis(in_features='polygonlyr',
                        out_feature_class=r'in_memory\polygon',
                        buffer_distance_or_field="{0} Meters".format(
                        bufferstart))

                    arcpy.Erase_analysis(in_features=r'in_memory\polygon',
                        erase_features='blyr',
                        out_feature_class=r'in_memory\clipbuffer')

                    polygon_this_area = [m[0] for m in arcpy.da.SearchCursor(
                        'in_memory\clipbuffer','SHAPE@AREA')][0]

                    print("Iteration BMP area = {0}".format(polygon_this_area))

####Bissection method variable change
                    if polygon_this_area < target_area_bmp:
                        bufferlower = bufferstart

                    elif polygon_this_area > target_area_bmp:
                        bufferupper = bufferstart

                    polygon_area = polygon_this_area

                    counter += 1
#After successfully finding the buffer value, evolves loop for feature counter
#and appends it to output feature
            feature_counter +=1
            gp.AddOutputsToMap = True
            arcpy.Append_management(inputs=r'in_memory\clipbuffer',
                target=output_feature_class, schema_type='NO_TEST')

            print('*************Feature {0} done*************'.format(
                feature_counter))
            counter += 1
#####END####

#Commands for mirror App
```

```
env = r'C:\Database.gdb'
Swales = r'C:\Default.gdb\swales_tomirror'
boundaries = r'C:\parcels'
output_feature_class = r'\swales_mirrored'
mirror(Swales,boundaries, output_feature_class)

#Commands for matchsize App
##########################################################################
mirrored_swales = r'C:\swales_mirrored'
output_feature_class_enlarged = r'\swales_mirrored_enlarged'
matchsize(mirrored_swales, boundaries, output_feature_class_enlarged)
```

ANNEX 8 – PySwatBmpApp-InfBasins Code

```
# -*- coding: utf-8 -*-
"""
Created on Sun Jan  6 16:41:59 2019

@author: david
"""
import archook
archook.get_arcpy()
import arcpy
import arcgisscripting, sys, traceback
arcpy.env.overwriteOutput = True
arcpy.CheckOutExtension("Spatial")

################    PySwatBmpApp-Swales  ##################################
"""The PySwatBmpApp-InfBasins Application is an ArcPy application for
(i)adjustinfbasin (a matchsize fuction) for in-parcel Infiltration Basins.

AUTHOR: DAVID BISPO FERREIRA(davidbispo@hotmail.com)
SUPPORT: FEDERAL UNIVERSITY OF PARANA(UFPR)/WATER RESOURCES AND ENVIRONMENTAL
ENGINEERING POST-GRADUATE PROGRAM(PPGERHA)/
INFRASTRUCTURE AND TRANSPORTATION TECHNOLOGY INSTITUTE(ITTI)

The adjustinfbasin function requires as INPUTS:

(i) a fileadress for an geodatabase ArcGIS environment
(i) a Infiltration Basins feature layer fileaddress As String
(ii) a street layout feature layer fileaddress As String
(iii) an output fileaddress As String

The adjustinfbasin function (a) fetches the parcel polygon area, using the
formulation proposed by Ferreira(2019),

(b)uses successive buffer+clip operations to reconstruct the target BMP area
within the parcel polygpon(since Infiltration Basins must be
inside a parcel according to the theoretical formulation of the program).

 *The mirror function returns as OUTPUTS:
A mirrored Infiltration Basin feature layer As ArcGIS Feature gdb feature layer
or shapefile
"""
def adjustinfbasin(env, inf_basins_feature, streets, output):

    arcpy.env.workspace=env
    #Input data#
    boundaries = streets
    polygons = inf_basins_feature
    out_feature_class = output


#Change to match your ok difference for area approximation
    ok_diff = 1
#creates the output feature
    arcpy.CreateFeatureclass_management(out_path=arcpy.env.workspace,
                                         out_name=out_feature_class,
                                        geometry_type='POLYGON',
                                        spatial_reference=arcpy.Describe(
                                                polygons).spatialReference)

#creates a boundary layer with a feature -> 'blyr'
    arcpy.MakeFeatureLayer_management(in_features=boundaries, out_layer='blyr')
```

```
    feature_counter = 1
#creates a cursor for the bmp polygon
    with arcpy.da.SearchCursor(polygons,['OID@','SHAPE@']) as cursor:
        for row in cursor:
            flag_deuruim = False
            iterationcounter = 0
            bufferlower = 0.
            bufferupper = 1000.

#query to select item
            sql = """{0} = {1}""".format(
                arcpy.AddFieldDelimiters(polygons,
                    arcpy.Describe(polygons).OIDFieldName),row[0])
            print(sql) #REMOVE IN RUNTIME

#creates temporary feature layer for the bmp layer ->'polygonlyr' w/ sql clause
            arcpy.MakeFeatureLayer_management(in_features=polygons,
                out_layer='polygonlyr',where_clause=sql)

#selects in the boundary layer the parcel that contains the bmp
            arcpy.SelectLayerByLocation_management(in_layer='blyr',
                overlap_type='CONTAINS', select_features='polygonlyr')
#fetches the area of the selected parcel
            try:
                boundary_area = [i[0] for i in arcpy.da.SearchCursor('blyr',
                    'SHAPE@AREA')][0]
                print("Boundary Area = {0}".format(boundary_area))

                bmp_area = [i[0] for i in arcpy.da.SearchCursor('polygonlyr',
                    'SHAPE@AREA')][0]

                print("BMP Area = {0}".format(bmp_area))

#conditionals for bmp sizing
                if boundary_area <= 5000:
                    max_area_for_bmp = boundary_area*0.04
                elif boundary_area > 5000 and boundary_area < 10000:
                    max_area_for_bmp = boundary_area*0.035
                elif boundary_area > 10000 and boundary_area < 20000:
                    max_area_for_bmp = boundary_area*0.030
                elif boundary_area > 20000 and boundary_area < 30000:
                    max_area_for_bmp = boundary_area*0.025
                elif boundary_area > 30000:
                    max_area_for_bmp = 4000

                print("Max BMP Area = {0}".format(max_area_for_bmp))

                wq_area_for_bmp = 0.4*bmp_area/0.003

                print("WQ Area = {0}".format(wq_area_for_bmp))

                if wq_area_for_bmp < max_area_for_bmp:
                    target_area_bmp = wq_area_for_bmp
                elif wq_area_for_bmp > max_area_for_bmp:
                    target_area_bmp = max_area_for_bmp

                print("Target area = {0}".format(target_area_bmp))
#########if final parcel area is smaller than the target area
                if [i[0] for i in arcpy.da.SearchCursor('polygonlyr',
                    'SHAPE@AREA')][0] < target_area_bmp:
```

```
polygon_area = [i[0] for i in arcpy.da.SearchCursor(
'polygonlyr','SHAPE@AREA')][0]

print("Starting BMP area = {0}".format(polygon_area))
print("Target BMP area = {0}".format(target_area_bmp))

bufferstart = (bufferlower + bufferupper)/2

while abs(polygon_area-target_area_bmp)>ok_diff:
    bufferstart = (bufferlower + bufferupper)/2
    print("Difference = {0}".format(abs(
            polygon_area-target_area_bmp)))

    print("Min = {0}, Max = {1}".format(bufferlower,
          bufferupper))
    print("Buffer = {0}".format(bufferstart))

    arcpy.Buffer_analysis(in_features='polygonlyr',

  out_feature_class=r'in_memory\polygon',
      buffer_distance_or_field="{0} Meters".format(
      bufferstart))

    arcpy.Clip_analysis(in_features=r'in_memory\polygon',
        clip_features='blyr',
        out_feature_class=r'in_memory\clipbuffer')

    polygon_area = [i[0] for i in arcpy.da.SearchCursor(
        r'in_memory\clipbuffer','SHAPE@AREA')][0]
    print("Iteration BMP area = {0}".format(polygon_area))

    if polygon_area < target_area_bmp:
        bufferlower = bufferstart

    elif polygon_area > target_area_bmp:
        bufferupper = bufferstart
feature_counter +=1
print("##########Feature {0} done#########".format(
      feature_counter))
arcpy.Append_management(inputs=r'in_memory\clipbuffer',
    target=out_feature_class, schema_type='NO_TEST')

########if final parcel area is larger than target area
        elif [i[0] for i in arcpy.da.SearchCursor('polygonlyr',
            'SHAPE@AREA')][0] > target_area_bmp:
            print("Parcel Area = {0}".format(boundary_area))

            polygon_area = [i[0] for i in arcpy.da.SearchCursor(
              'polygonlyr','SHAPE@AREA')][0]

            print("Starting BMP area = {0}".format(polygon_area))
            print("Target BMP area = {0}".format(target_area_bmp))
            bufferlower = bufferupper *-1
            bufferupper = 0
            bufferstart = ((bufferlower + bufferupper)/2)

            while abs(polygon_area-target_area_bmp)>ok_diff:

                    bufferstart = (bufferlower + bufferupper)/2
                    print("Difference = {0}".format(
```

```
                        polygon_area-target_area_bmp))

                    print("Min = {0}, Max = {1}".format(bufferlower,
                        bufferupper))
                    print("Buffer = {0}".format(bufferstart))

                    arcpy.Buffer_analysis(in_features='polygonlyr',
                        out_feature_class=r'in_memory\polygon',
                        buffer_distance_or_field="{0} Meters".format(
                            bufferstart))

                    arcpy.Clip_analysis(
                        in_features=r'in_memory\polygon',
                        clip_features='blyr',
                            out_feature_class=r'in_memory\clipbuffer')
                    try:
                        polygon_area = [i[0] for i \n
                        in arcpy.da.SearchCursor(
                        r'in_memory\clipbuffer','SHAPE@AREA')][0]

                        print("Iteration BMP area = {0}".format(
                                polygon_area))

                        if polygon_area > target_area_bmp:
                            bufferupper = bufferstart
                        elif polygon_area < target_area_bmp:
                            bufferlower = bufferstart
                    except:
                        print("Buffer Failed. Reducing buffer...")
                        if bufferstart <=-100:
                            bufferlower += 50
                        elif bufferstart >-100 and bufferstart <=-50:
                            bufferlower += 10
                        elif bufferstart >-50 and bufferstart <=-10:
                            bufferlower += 5
                        elif bufferstart > -10:
                            bufferlower += 0.25
                    iterationcounter+=1
                    if iterationcounter >70:
                        flag_deuruim = True
                        break
                        print("Deu ruim! Pulando...")
                if flag_deuruim == False:
                    feature_counter +=1
                    print("########Feature {0} done ###########".format(
                        feature_counter))
                    arcpy.Append_management(inputs=r'in_memory\clipbuffer',
                        target=out_feature_class, schema_type='NO_TEST')
                else:
                    pass
        except:
            print("&*&*&*&Deleting object -> {0} &*&*&".format(sql))
            pass
    print("Finished!")

env = r'C:\database.gdb'
InfBasins = r'C:\database.gdb\inf_basins'

boundaries = r'C:\database.gdb\lotes_basin_swat_clip'
out_feature_class = r'\inf_basins_done'
```

```
adjustinfbasin(env, InfBasins, boundaries, out_feature_class)
```

ANNEX 9 PySwatBmpApp-InfBasins Code

```
# -*- coding: utf-8 -*-
"""
Created on Sun Jun 10 21:25:42 2018

@author: david
"""

import os
import dic_par
import listtype
import datetime
from distutils.dir_util import copy_tree
class par_handler:

    def __init__(self,TXTInOut):

        self.TXTInOut = TXTInOut

    def inforeader(self, parser):
#Information reader
        FirstLine = parser[0]
        firstline_read_single_list = list(FirstLine)
        check_sub = FirstLine.find("sub")
        check_rte = FirstLine.find("rte")

        if check_sub != -1 or check_rte != -1:
            HRU_number = None
            LULC_type = None
            FirstLineSplit = FirstLine.split()
            SubBasinPos = FirstLineSplit.index('Subbasin:')
            sub_number = int(FirstLineSplit[SubBasinPos+1])

            return LULC_type, sub_number,HRU_number

        else:
            SubBasinPos = FirstLine.find('Subbasin:')
            SubBasinStart = SubBasinPos + 9

            HRUPos = FirstLine.find('HRU:',26)
            SubBasinEnd = HRUPos + -1


            LULCPos = FirstLine.find('Luse:')
            LULCStart = LULCPos+5
            LULCEnd = LULCStart + 3
            LULC_scan = firstline_read_single_list[LULCStart:LULCEnd+1]
            LULC_type = ''.join(LULC_scan)


            sub_number = firstline_read_single_list[SubBasinStart:
                SubBasinEnd+1]

            sub_number = ''.join(sub_number)
            sub_number = int(sub_number)

            HRU_abs_Pos = FirstLine.find('HRU:')
            HRU_abs_Start = HRU_abs_Pos+4
            HRU_abs_End = SubBasinPos-1
            HRU_scan = firstline_read_single_list[HRU_abs_Start:HRU_abs_End+1]
            HRU_scan = ''.join(HRU_scan)
```

```
            HRU_number = HRU_scan.replace(" ", "")
            HRU_number = int(HRU_number)

            return LULC_type, sub_number,HRU_number

    def change(self, parameter, method, value, sb = 'all' , lulc = 'all',
            hru = None, log= r'E:\log.txt'):

        def log_end(parameter,method,value,files_done,total_files):
            print("Writing to log file...")
            if os.path.isfile(log) == True:
                openfile = open(log, 'r')
                lines = openfile.readlines()
                len_lines = len(lines)
                openfile.close()

                timenow = datetime.datetime.now()
                line_to_append = ("\n"+ str([len_lines,parameter,method,value,
                    timenow]))

                spamwriter = open(log, 'a+')
                spamwriter.write(line_to_append)
                spamwriter.close()

            else:
                choice = input("You don't have a logfile. Do you want to\
                               create one? [y/n]: ")
                if choice == "y":
                    spamwriter = open(log, 'w')
                    timenow = datetime.datetime.now()
                    spamwriter.write(str([0,parameter,method,value,timenow]))
                    spamwriter.close()
                    if choice == "n":
                        pass
                        print("Warning! You did not print this change to the\
                            log!")
            print ('Program complete! -> %.0f files altered in a total of %.0f'
                   %(files_done,total_files))
            print ('You have successfully changed %s %s with a value of %s'
                   %(parameter,method,value))
        print("###Starting parameter changer...")
#Locates the file extension and line number in a dictionary

        instance = dic_par.param_dic()
        target_file, linenumber, exceptions = instance.dic_query(parameter)
#gets a filelist for the specificed folder and format
        soil_filenumber = len(listtype.listtype(self.TXTInOut,".sol"))
        sub_filenumber = len(listtype.listtype(self.TXTInOut,".sub"))-1
        #minus 1: output.sub
        print ("You have %.0f hrus and %.0f subbasins this project" %
            (soil_filenumber,sub_filenumber))

        filelist = listtype.listtype(self.TXTInOut,target_file)
#Uses the fetched list to change values
        total_files = len(filelist)
        files_done = 0

        for exception in exceptions:
            flag = exception in filelist
```

```
            if flag == True:#checks if output file has indeed been printed
                index_exception = filelist.index(exception)
                del filelist[index_exception]
#####Routine for sol files      ####
        if target_file == '.sol':
            for filename in filelist:
                address = os.path.join(self.TXTInOut, filename)
                openfile = open(address, 'r')
                try:
                    parser = openfile.readlines()
                    openfile.close()
                    LULC_type, sub_number, hru_number = self.inforeader(parser)
                except:
                    print ('Your file %s has problems') % filename
                    exit()

                if hru == None and lulc == 'all'and (sb == sub_number or
                                                  sub_number in sb)\
                or hru == None and (lulc == 'all') and (sb == 'all')\
                or hru == None and (lulc == LULC_type or LULC_type in lulc)\
                    and sb == 'all'\
                or hru == None and (lulc == LULC_type or LULC_type in lulc)\
                    and (sb == sub_number or sub_number in sb)\
                or (hru != None and type(hru) == int and  hru_number == hru)\
                or (hru != None and type(hru) == list and (hru_number in hru or hru_number == hr

                    line = parser[linenumber]


                    linesplit = line.split()
                    #linesplit_data_str = linesplit[3:]
                    list_line = list(line)
                    if parameter == 'SOL_AWC':
                        linesplit_data = linesplit[6:]
                    elif parameter == 'SOL_K':
                        linesplit_data = linesplit[3:]

                    for i in range(len(linesplit_data)):
                        linesplit_data[i] = float(linesplit_data[i])
#Replace
                    if method == 'replace':
                        print('Dont redo replace in soil files, you will screw\
                            your model! Exiting program!')
                        break
                        exit()
#Relative
                    if method == 'relative':
#Gets a parameter list from the backup folder
                        folder_backup = os.path.join(self.TXTInOut, 'Backup')
                        address_backup = os.path.join(folder_backup, filename)

                        if os.path.isdir(folder_backup) == False:
                            os.makedirs(address_backup)
                            print('Have a backup folder set up before starting')
                            exit()
                            #copy_tree(address, address_backup)
                        elif os.path.isdir(folder_backup) == True:

                            openfile_bkp = open(address_backup, 'r')
                            parser_bkp = openfile_bkp.readlines()
```

```
                    openfile_bkp.close()
                    try:
                       line_bkp = parser_bkp[linenumber]
                       linesplit_bkp = line_bkp.split()
                       if parameter == 'SOL_AWC':
                           linesplit_data_bkp = linesplit_bkp[6:]

                       elif parameter == 'SOL_K':
                           linesplit_data_bkp = linesplit_bkp[3:]

                    except:
                         print("Your file %s has problems"
                               %address_backup)
#Multiplies all values in list by the desired value
                    for i in range(len(list(linesplit_data_bkp))):
                       linesplit_data_bkp[i] = (
                               float(linesplit_data_bkp[i]))*(1+value)
#Finds the end and beggining of string
                    for k in range(len(linesplit_data_bkp)):
                        end_old = 38 + 12* k
                        start_old = end_old - len(
                                str(linesplit_data[k]))+1
#Replaces the string positions
                        for j in range(start_old,end_old+1):
                            list_line[j] = ''
                        end_new = end_old
                        string = '%.3f'%(linesplit_data_bkp[k])
                        startwrite = end_new - len(string)+1
                        #spaces = len(string) # Variable inspection
                        list_line[startwrite:end_new+1] = list(string)
                    list_line.insert(39, '')
                    list_line = ''.join(list_line)
                    parser[linenumber] = list_line
#Writes output file
                    spamwriter = open(address, 'w')
                    for line in parser:
                        spamwriter.write(line)
                    spamwriter.close()
                    files_done +=1
            log_end(parameter,method,value,files_done,total_files)
#####Routine for non-sol files####
        elif target_file != '.sol':
#Open file
            for filename in filelist:
                address = os.path.join(self.TXTInOut, filename)
                openfile = open(address, 'r')
                parser = openfile.readlines()
                openfile.close()
                LULC_type, sub_number, hru_number = self.inforeader(parser)

#Routine for rte or sub files
                if target_file == '.rte' or target_file == '.sub':

                    if sub_number == sb or sub_number in sb:

                        try:
                            line = parser[linenumber]
                        except:
                            print ('Your file %s has problems') % filename
                            exit()
```

```
            if method == 'replace':

                float_replace = float(value)
                str_replace = '%.3f' % float_replace
                list_replace = list(str_replace)
#Position calculator
                list_line = list(line)
                list_other_line = list(parser[4])
                first_position = 15 - len(list_replace)+1

                for i in range(16):
                    list_line[i] = ' '

                list_line[first_position:16] = list_replace
                list_line_to_str = ''.join(list_line)
                if parameter == 'CH_N1' or parameter == parameter\
                    == 'ALPHA_BNK' or parameter=='CH_K2':
                    list_line_to_str = list_line_to_str.replace(
                            " ", "", 2)
                    list_line_to_str = list_line_to_str.replace(
                            "|", " |", 1)
                    list_line_to_str = list_line_to_str.replace(
                            "|", " |", 1)

                elif parameter == 'CH_N2':
                    pass

                parser[linenumber] = list_line_to_str
                spamwriter = open(address, 'w')
                for linhe in parser:
                    spamwriter.write(linhe)
                spamwriter.close()
                files_done +=1
#relative method
            elif method == 'relative':

#Parses the File in the Backup folder for value
                folder_backup = os.path.join(self.TXTInOut,
                                             'Backup')
                address_backup = os.path.join(folder_backup,
                                              filename)

                if os.path.isdir(folder_backup) == False:
                    os.makedirs(address_backup)
                    copy_tree(address, address_backup)
                elif os.path.isdir(folder_backup) == True:

                    openfile = open(address_backup, 'r')
                    parser = openfile.readlines()
                    try:
                        line_bkp = parser[linenumber]
                        par_orig_value = line_bkp[:16]
                        par_orig_value = float(par_orig_value)
                        if par_orig_value == 0:
                            print("Your file file %s has a starting\
                                value for %s of zero. Please check\
                                for relative references"%
                                (filename, parameter))
                        par_new_value = par_orig_value * (1+value)
```

```
                                    openfile.close()
                                except:
                                    print ('Your file %s has problems') % filename
                                    exit()
#Replaces the parameter with a new calculated value
                                str_replace = '%.3f' % par_new_value
                                list_replace = list(str_replace)
#Position calculator
                                line = parser[linenumber]
                                list_line = list(line)

                                if parameter == 'CH_N2':
                                    first_position = 14 - len(list_replace)+1
                                    final_position = 14
                                else:
                                    first_position = 16 - len(list_replace)+1
                                    final_position = 16
#Clears all spaces in line as list from file

                                if parameter == 'CH_N2':
                                    erase=17
                                else:
                                    erase=20
                                for i in range(erase):
                                    list_line[i] = ' '
                                j = 0

                                for k in range(first_position-1,
                                               final_position):
                                    list_line[k] = list_replace[j]
                                    j +=1

                                list_line_to_str = ''.join(list_line)
                                parser[linenumber] = list_line_to_str

                                spamwriter = open(address, 'w')
                                for linesa in parser:
                                    spamwriter.write(linesa)
                                spamwriter.close()
                                files_done +=1
                else:#CHECKER FOR NON SUBBASIN FILES
#Condition checker
                    if hru == None and lulc == 'all' and (str(sb) == str(
                        sub_number) or type(sb) == list and sub_number in sb)\
                    or hru == None and (lulc == 'all') and (sb == 'all')\
                    or hru == None and (lulc == LULC_type or LULC_type in lulc)\
                        and sb == 'all'\
                    or hru == None and (lulc == LULC_type or LULC_type in lulc)\
                        and (str(sb) == str(sub_number) or sub_number in sb)\
                    or (hru != None and type(hru) == int and  hru_number == hru)\
                    or (hru != None and type(hru) == list and (hru_number in\
                        hru or hru_number == hru)):

                        try:
                            line = parser[linenumber]
                            openfile.close()
                        except:
                            print ('Your file %s has problems') % filename
                            exit()
#replace method
```

```python
if method == 'replace':

    float_replace = float(value)
    str_replace = '%.3f' % float_replace
    list_replace = list(str_replace)
#Position calculator
    list_line = list(line)
    list_other_line = list(parser[4])
    first_position = 15 - len(list_replace)+1

    for i in range(16):
        list_line[i] = ' '

    list_line[first_position:16] = list_replace

    list_line_to_str = ''.join(list_line)
    parser[linenumber] = list_line_to_str

    spamwriter = open(address, 'w')
    for line_unit in parser:
        spamwriter.write(line_unit)
    spamwriter.close()
    files_done +=1

#relative method
elif method == 'relative':

#Parses the file in the Backup folder for value
    folder_backup = os.path.join(self.TXTInOut,
                                 'Backup')
    address_backup = os.path.join(folder_backup, filename)

    if os.path.isdir(folder_backup) == False:
        os.makedirs(address_backup)
        copy_tree(address, address_backup)
    elif os.path.isdir(folder_backup) == True:
        try:
            openfile = open(address_backup, 'r')
            parser_bkp = openfile.readlines()
            openfile.close()
        except:
            print ('Your backup file %s has problems\
                   and could not be opened') % filename
            exit()
        try:
            line_bkp = parser_bkp[linenumber]
            par_orig_value = line_bkp[:16]
            par_orig_value = float(par_orig_value)
            par_new_value = par_orig_value * (1+value)

        except:
            print ('The parsing operation on file %  \
                   could not be performed') % filename
            exit()
#Replaces the parameter with a new calculated value
        str_replace = '%.3f' % par_new_value
        list_replace = list(str_replace)
#Position calculator
        line = parser[linenumber]
        list_line = list(line)
```

```
                                first_position = 16 - len(list_replace)+1

#Clears all spaces in line as list from file
                                for i in range(20):
                                    list_line[i] = ' '
                                j = 0
                                for k in range(first_position-1, 16):
                                    list_line[k] = list_replace[j]
                                    j +=1

                                list_line_to_str = ''.join(list_line)
                                parser[linenumber] = list_line_to_str

                                spamwriter = open(address, 'w')
                                for line_units in parser:
                                    spamwriter.write(line_units)
                                spamwriter.close()
                                files_done +=1
        log_end(parameter,method,value,files_done,total_files)

##Execution command example##

TXTInOut = r'C:\TxtInOut'
instance = par_handler(TXTInOut)
fileadress = os.path.join(TXTInOut,'output.rch')
variables = ["FLOW_OUTcms, SEDYLD, NTOT"]

instance.change(parameter = 'CN2', method = 'relative', value = -0.083233,
                sb=1,log =r'D:\log.txt')
instance.change(parameter = 'CN2', method = 'relative', value = -0.083233,
                sb=list(range(1,40)),log = r'D:\log.txt')
```

ANNEX 10 – PySwatGetIO-change_par Code - APPLICATION FOR CALIBRATION

```
# -*- coding: utf-8 -*-
"""
Created on Tue Dec 11 22:53:45 2018

@author: David Bispo
"""
import pySWATParApp as Ppar
import get_par as gp
import get_output as go
import os
import subprocess
from subprocess import Popen, PIPE
import numpy as np

def execute(cmd):# Runs SWAT into the current terminal
    popen = subprocess.Popen(cmd, stdout=subprocess.PIPE,
                             universal_newlines=True)
    for stdout_line in iter(popen.stdout.readline, ""):
        yield stdout_line
    popen.stdout.close()
    return_code = popen.wait()
    if return_code:
        raise subprocess.CalledProcessError(return_code, cmd)

#TXTInOut = r'C:\Barigui_Biorretention\Scenarios\Calibrated\TxtInOut'
TXTInOut = r'C:\Barigui_Biorretention\Scenarios\Altered\TxtInOut'
instance_Ppar = Ppar.par_handler(TXTInOut)
fileadress = os.path.join(TXTInOut,'output.rch')
variables = ["FLOW_OUTcms"]
observed = r'E:\OneDrive\master_spreadsheet.csv'
rel_table = {"tmd":39, "stq":98, "cax":155}

###COMANDS FOR SIMULATION CALIBRATION
instance_Ppar.change(parameter = 'GWQMN', method = 'replace', value = 1850.233,
                     sb= list(range(1,40)))#2
instance_Ppar.change(parameter = 'RCHRG_DP', method = 'replace', value =0.9503,
                     sb= list(range(1,40)))#3
instance_Ppar.change(parameter = 'HRU_SLP', method = 'relative', value =-0.206,
                     sb= list(range(1,40)))#4
instance_Ppar.change(parameter = 'SLSOIL', method = 'replace', value = -4.483,
                     sb= list(range(1,40)))#4
instance_Ppar.change(parameter = 'OV_N', method = 'relative', value = 0.000780,
                     sb= list(range(1,40)))#5#
instance_Ppar.change(parameter = 'LAT_TTIME', method = 'replace', value =16.915,
                     sb= list(range(1,40)))#6
instance_Ppar.change(parameter = 'CH_N1', method = 'relative', value =-0.0138 ,
                     sb= list(range(100,155)))#14
instance_Ppar.change(parameter = 'CH_K1', method = 'replace', value = 466.174,
                     sb= list(range(1,40)))#7
instance_Ppar.change(parameter = 'ALPHA_BNK', method = 'replace', value =0.211,
                     sb= list(range(1,40)))#8
instance_Ppar.change(parameter = 'CH_K2', method = 'replace', value = 373.52,
                     sb= list(range(1,40)))#9
instance_Ppar.change(parameter = 'SOL_AWC', method = 'relative', value = 0.378,
                     sb= list(range(1,40)))#10
instance_Ppar.change(parameter = 'SOL_K', method = 'relative', value = -0.827,
                     sb= list(range(1,40)))#10


instance_Ppar.change(parameter = 'CN2', method = 'relative', value = -0.164,
```

1

```
                   sb= list(range(40,99)))#1
instance_Ppar.change(parameter = 'RCHRG_DP', method = 'replace',value=0.277443,
                   sb= list(range(40,99)))#12
instance_Ppar.change(parameter = 'LAT_TTIME', method = 'replace', value= 4.365,
                   sb= list(range(40,99)))#13
instance_Ppar.change(parameter = 'CH_K1', method = 'replace', value = -45.366,
                   sb= list(range(40,99)))#14
instance_Ppar.change(parameter = 'CH_N2', method = 'relative', value =  -0.332,
                   sb= list(range(40,99)))#14
instance_Ppar.change(parameter = 'ALPHA_BNK', method = 'replace', value= 0.794,
                   sb= list(range(40,99)))#15
instance_Ppar.change(parameter = 'CH_K2', method = 'replace', value = 216.269,
                   sb= list(range(40,99)))#25
instance_Ppar.change(parameter = 'SOL_AWC', method = 'relative', value = 0.564,
                   sb= list(range(40,99)))#16


instance_Ppar.change(parameter = 'CN2', method = 'relative', value = 0.0364,
                   sb= list(range(100,155)))#1
instance_Ppar.change(parameter = 'ALPHA_BF', method = 'replace', value = 1.013,
                   sb= list(range(100,155)))#1
instance_Ppar.change(parameter = 'GWQMN', method = 'replace', value = 4178.194,
                   sb= list(range(100,155)))#20
instance_Ppar.change(parameter = 'RCHRG_DP', method = 'replace', value = 0.943,
                   sb= list(range(100,155)))#21
instance_Ppar.change(parameter = 'HRU_SLP', method = 'relative', value =-0.188,
                   sb= list(range(100,155)))#4
instance_Ppar.change(parameter = 'LAT_TTIME', method = 'replace', value =14.481,
                   sb= list(range(100,155)))#13
instance_cp.change(parameter = 'CH_N1', method = 'relative', value =  0.134,
                   sb= list(range(100,155)))#14
instance_cp.change(parameter = 'CH_N2', method = 'relative', value = -0.0346,
                   sb= list(range(100,155)))#232
instance_cp.change(parameter = 'ALPHA_BNK', method = 'replace', value = 0.654,
                   sb= list(range(100,155)))#24
instance_cp.change(parameter = 'CH_K2', method = 'replace', value = 175.229,
                   sb= list(range(100,155)))#25

os.chdir(r'E:\Barigui_InfBasins\TxtInOut')        ###RUN SWAT CALL
print("Running Swat..")
for path in execute(["swat.exe"]):
    print(path), end="")

instance_go = go.output_rch(fileadress,[39,98,155],variables,#Reads SWAT OUTPUT
                            observed_input = observed, rel_table = rel_table)

#getting all reaches data
array = instance_go.read_rch()

print('Writing to simulations log file...')#WRITES SIMULATIONS TO LOG FILE

output_sims = r'E:\sim_result_InfBasins.txt'
if os.path.isfile(output_sims) == False:
    np.savetxt(output_sims,array, fmt='%s')
else:
    temp = np.loadtxt(output_sims, dtype = str)
    array_to_stack = array[:,2:3]
    temp = np.concatenate((temp,array_to_stack), axis=1)
    np.savetxt(output_sims,temp,fmt='%s')
"""
```

```
#CHANGES CN VALUES FOR CERTAIN LAND USES - NOT ACTIVE IN THIS CODE

print('changing parameters on Biorretention...')
instance_Ppar.change(parameter = 'CN2', method = 'relative', value = -0.5, sb= 'all', lulc ='BR
instance_Ppar.change(parameter = 'CN2', method = 'relative', value = -0.5, sb= 'all', lulc ='IB

"""
print('Programa terminado!!!!!')
"""
```

ANNEX 11 – PySwatGetIO-get_par

```
# -*- coding: utf-8 -*-
"""
Created on Sun Sep 23 17:28:50 2018

@author: david
"""
import numpy as np
import dic_par #importing the dictionary code
import listtype
import os

class swat_partable:

    def __init__(self, TXTInOut,sb, lulc):
        self.TXTInOut,self.sb, self.lulc, = TXTInOut, sb, lulc

    def get_par_hru(self):

        dic_instance = dic_par.param_dic() #opening an instance of dic
        keys = dic_instance.dic_keys() # fetching the keys
        if 'SOL_K' in keys:
            x = keys.index('SOL_K')
            del keys[x]
        if 'CH_K1' in keys:
            x = keys.index('CH_K1')
            del keys[x]
        if 'ALPHA_BNK' in keys:
            x = keys.index('ALPHA_BNK')
            del keys[x]
        if 'CH_K2' in keys:
            x = keys.index('CH_K2')
            del keys[x]
        if 'SOL_AWC' in keys:
            x = keys.index('SOL_AWC')
            del keys[x]
        if 'CH_N2' in keys:
            x = keys.index('CH_N2')
            del keys[x]

        TXTInOut, sb, lulc = self.TXTInOut, self.sb, self.lulc

        print("Starting hru parameter fetcher...")
#gets a filelist for the specificed folder and format
        soil_filenumber = len(listtype.listtype(TXTInOut,".sol"))
        sub_filenumber = len(listtype.listtype(TXTInOut,".sub"))-1
        #minus 1: output.sub
        print ("You have %.0f hrus and %.0f subbasins this project" %
           (soil_filenumber,sub_filenumber))

        for k in keys:
            target_file,line,exceptions = dic_instance.dic_query(k)
            filelist = listtype.listtype(TXTInOut,target_file)

            for exception in exceptions:
                flag = exception in filelist
                if flag == True:#checks if output file has indeed been printed
                    index_exception = filelist.index(exception)
                    del filelist[index_exception]

            for i in filelist:
```

```
#gets the paramater data in file
adress = os.path.join(TXTInOut, i)
openfile = open(adress, 'r')
parser = openfile.readlines()

FirstLine = parser[0]
firstline_read_single_list = list(FirstLine)

SubBasinPos = FirstLine.find('Subbasin:')
SubBasinStart = SubBasinPos + 9

HRUPos = FirstLine.find('HRU:')
HRU_Start = HRUPos+4
HRU_End = SubBasinPos - 1

HRU_number = firstline_read_single_list[HRU_Start:HRU_End+1]
HRU_number = ''.join(HRU_number)
try :
    HRU_number = int(HRU_number)
except:
    print (('Failed to red HRU number on file %s, key %s') %
           (i, k))
    exit()

SecondHRUPos = FirstLine.find('HRU:',26)

SubBasinEnd = SecondHRUPos + -1

LULCStart = FirstLine.find('Luse:')+5
LULCEnd = LULCStart + 3
LULC_scan = firstline_read_single_list[LULCStart:LULCEnd+1]
LULC_type = ''.join(LULC_scan)

SecondHRU_Start = SecondHRUPos+4
SecondHRU_End = FirstLine.find('Luse:') - 1

SecondHRU_number = firstline_read_single_list[SecondHRU_Start:
    SecondHRU_End+1]
SecondHRU_number = ''.join(SecondHRU_number)
SecondHRU_number = int(SecondHRU_number)

sub_number = firstline_read_single_list[SubBasinStart:
    SubBasinEnd+1]

sub_number = ''.join(sub_number)
sub_number = int(sub_number)


    #inputs a single or all subbasins and lulcs
parsed = parser[line]
parameter = list(parsed)
parameter = parameter[:16]
parameter = ''.join(parameter)
try :
    parameter = float(parameter)
except:
    print (('Failed to convert to float on file %s, key %s' %
            (i, k)))
    exit()
openfile.close()
```

```
                if k == keys[0]:

                    if i == filelist[0]:
                        par_name = k
                        a = np.array(["LULC","Sub_number","HRU_Number_ABS",
                                      "HRU_Number_REL", par_name]).T
                        new_table = np.array([LULC_type,sub_number,HRU_number,
                                              SecondHRU_number, parameter])
                        a = np.vstack((a,new_table))
                    else:
                        new_table = np.array([LULC_type,sub_number,HRU_number,
                                              SecondHRU_number, parameter])
                        a = np.vstack((a,new_table))
                else:
                    if i == filelist[0]:
                        par_name = k
                        array = np.array([par_name,parameter]).T

                    elif i != filelist[0] and i != filelist[-1]:
                        array = np.append(array,parameter)

                    elif i == filelist [-1] and k != keys[-1] :
                        array = np.append(array,parameter)
                        a = np.column_stack((a,array))

                    elif i == filelist [-1] and k == keys[-1] :
                        array = np.append(array,parameter)
                        a = np.column_stack((a,array))
                        print("Concluido")
                    elif k == keys[-1] and i == filelist[-1]:
                        a = np.column_stack((a,array))
###Filters by Land use and subbasin number
#Start - Gets data and filters in header and body

        a_header = a[0,:].T
        a_body = a[1:,:]
        lulc_col = int(np.where(a_header == 'LULC')[0][0])
        sb_col = int(np.where(a_header == 'Sub_number')[0][0])

        if type(sb) == int:
            a_body = a_body[a_body[:,sb_col] == str(sb)]

        elif type(sb) == list:
            for o in sb:
                if len(sb) == 1 and o == sb[0]:
                    a_body_filtered = a_body[a_body[:,sb_col] == str(o)]
                    a_body = a_body_filtered
                elif len(sb) != 1 and o == sb[0]:
                    a_body_filtered = a_body[a_body[:,sb_col] == str(o)]
                elif len(sb) != 1 and o != sb[0] and o != sb[-1]:
                    new_a_body_filtered = a_body[a_body[:,sb_col] == str(o)]
                    a_body_filtered = np.vstack((a_body_filtered,
                                                 new_a_body_filtered))
                elif len(sb) != 1 and o == sb[-1]:
                    new_a_body_filtered = a_body[a_body[:,sb_col] == str(o)]
                    a_body_filtered = np.vstack((a_body_filtered,
                                                 new_a_body_filtered))
                    a_body = a_body_filtered

        elif sb == 'all':
```

```
            pass

        if type(lulc) == str and lulc != 'all':
            a_body = a_body[a_body[:,0] == str(lulc)]

        elif type(lulc) == list:
            for p in lulc:
                if p == lulc[0]:
                    a_body_filtered = a_body[a_body[:,lulc_col] == str(p)]
                    if len(lulc) == 1:
                        a_body = a_body_filtered
                elif p != lulc[0] and p != lulc[-1]:
                    new_a_body_filtered = a_body[a_body[:,lulc_col] == str(p)]
                    a_body_filtered = np.vstack((a_body_filtered,
                                                 new_a_body_filtered))
                elif p == lulc[-1]:
                    new_a_body_filtered = a_body[a_body[:,lulc_col] == str(p)]
                    a_body_filtered = np.vstack((a_body_filtered,
                                                 new_a_body_filtered))
                    a_body = a_body_filtered

        filtered_table = np.vstack((a_header,a_body))
        full_table = a
        return filtered_table, full_table
        print("Done!")

####EXAMPLE CODE RUN
input_folder = r'C:\Barigui_InfBasins\Scenarios\Altered\TxtInOut'
instance = swat_partable(TXTInOut = input_folder, sb = 95, lulc='all')
table, original_table = instance.get_par_hru()
```

ANNEX 12 – PySwatGetIO-get_output Code

```
# -*- coding: utf-8 -*-
"""
Created on Sun Aug 19 21:16:30 2018

@author: david
"""
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import dic_par
import datetime

def NS(s,o):
    """
    Nash Sutcliffe efficiency coefficient
    input:
        s: simulated
        o: observed
    output:
        ns: Nash Sutcliffe efficient coefficient
    """
    return 1 - sum((s-o)**2)/sum((o-np.mean(o))**2)
class output_rch:

    def __init__(self,fileaddress,rch_filter,par_filter,observed_input = None,
                 rel_table = None):
        self.rch_filter = rch_filter
        self.observed_input = observed_input
        self.rel_table = rel_table
        self.par_filter = par_filter
        self.fileaddress = fileaddress

    """Takes an output.rch file and parses it for reach number and desired
    parameters
    fileadress = the file address as string
    rch_filter = takes integer, list of integers and 'all' as arguments
    par_silter = takes string, list of strings and 'all' as arguments
    observed_input = Numpy array with header: "GDAY", "STATION1_FLOW",
    "STATION2_FLOW", ..."STATIONn_FLOW"
    rel_table = nx2 array with first collumn "RCH_NUMBER", "STATION NAME"

    returns numpy array

    Methods: read_rch(), print_allinone(output_folder),
    print_manyinfolder(output_folder)
    """

    def getns(self):
        if self.observed_input == None:
            print("Your observed series file is empty")
        else:
            nrows_series = temp_array_date.shape[0]
            keys = list(rel_table.keys())
            ns_dic = {}
            for i in keys:
                key_from_rch = rel_table[i]
                if key_from_rch in rch_filter:
                    series = mydata[i]
                    series_array = series.values
                    nrows_mydata = series_array.shape[0]
```

```
                ns = NS(eff_array[:,0],eff_array[:,1])
        print (ns_dic)
        return ns_dic

    def read_rch(self):
#Function local parameter definition
        rch_filter = self.rch_filter
        par_filter = self.par_filter
        observed_input = self.observed_input
        rel_table = self.rel_table
#Reads personal data
        if observed_input != None and rel_table != None:
            self.my_data = pd.read_csv(observed_input, delimiter=',',
                                       index_col = "GDAY")
            self.flag_pdata = True
        else:
            print("Please input both observed inputs and relational tables
                  correctly!")
#Opens file and gets header
        from_file = open(self.fileaddress, 'r')
        lines = from_file.readlines()
        result_list=lines[9:]
        header = lines[8].split()
#Creates list of lists for result array array
        result_list_parse1 = []
        for i in result_list:
            a = i.split()[1:]
            result_list_parse1.append(a)
#Creates array with the result array list of lists
        #Creates array from only data array
        result_array = np.array(result_list_parse1)
        #tests for printed calendar dates
        if ("MO") in header:

            mo_address = header.index("MO")# Address for gregorian day
            day_address = header.index("DA")
            year_address = header.index("YR")

        else:
            mo_address = header.index("MON")

        rch_address = header.index("RCH")#Address for rch collumn

#Parses data from the rch_filter parameter - Takes integer and list or all###
        if rch_filter != 'all':
            if type(rch_filter) == int :
                result_array_filtered=result_array[result_array[:,
                                            rch_address] == str(rch_filter)]
            elif type(rch_filter) == list :
                for k in rch_filter:
                    if k == rch_filter[0]:
                        array_filtered = result_array[result_array[:,
                                            rch_address] == str(k)]
                    else:
                        array_filtered_temp = result_array[result_array[:,
                                                  rch_address] == str(k)]
                        array_filtered = np.vstack((array_filtered,
                                                array_filtered_temp))

                result_array_filtered = array_filtered
```

```
        if ("MO") in header:

            dayArray = result_array_filtered[:,day_address]
            monthArray = result_array_filtered[:,mo_address]
            yearArray = result_array_filtered[:,year_address]

            date_zero = (datetime.date(int(yearArray[0]),int(monthArray[0]),
                                       int(dayArray[0])))
            gdays = []
            date_list = []
            date_list_string = []

            for i in range(result_array_filtered.shape[0]):
                date_temp = (datetime.date(int(yearArray[i]),int(monthArray[i]),
                                           int(dayArray[i])))
                gdays.append((date_temp - date_zero).days)
                date_list.append(date_temp)
                date_list_string.append(date_temp.strftime('%m/%d/%Y'))

            date_array = np.array(gdays)
            days_sim = date_array
            result_array_filtered = np.insert(result_array_filtered,2,
                                    days_sim,axis = 1) #inserts gday simday array
            header.insert(2,"SDG")
            #inserts on the header the Simulation Gregorian days Collumn header

        else:
            gdays = list(range(1,result_array_filtered.shape[0]+1))
            date_array = np.array(gdays)
            days_sim = date_array
            result_array_filtered = np.insert(result_array_filtered,2,
                                  days_sim,axis = 1) #inserts gday simday array
            header.insert(2,"SDG")
            #inserts on the header the Simulation Gregorian days Collumn header
#Finds the number of collumns to be fetched
        if type(par_filter) == list:
            coln_final_array = len(par_filter)
        if type(par_filter) == str:
            coln_final_array = 1

        a = np.empty((result_array_filtered.shape[0], int(coln_final_array)))
        gdays_row = np.array(result_array_filtered[:,2],dtype = int)
        a = np.insert(a,0,gdays_row, axis = 1)
        rch_row = np.array(result_array_filtered[:,0],dtype = int)
        a = np.insert(a,1,rch_row, axis = 1)
        #date_row = np.array(date_list_string)
        #a = np.insert(a,1,date_row, axis = 1)
        col_index_list = []
#Parses and filters the par_filter parameter - Takes string, list of strings as
    #parameters and 'all'
        if par_filter != 'all':
            if type(par_filter) == list:
                for i in par_filter:
    #Fetches the parameter filter as a collumn number
                    col_interest = header.index(i)
                    col_index_list.append(col_interest)
            if type(par_filter) == str:
    #Fetches the parameter filter as multiple collumn numbers
                col_interest = header.index(par_filter)
```

```
                col_index_list.append(col_interest)
#Creates the final array with the desired variables
        count = 2
        for i in col_index_list:
            analysis_array = result_array_filtered[:,i]
            a[:,count] = analysis_array
            count = count+1

        if type(par_filter) == list:
            a_header = ["GDAY", "RCH"]
            for i in par_filter:
                a_header.append(i)
            a_header_array = np.array(a_header).T
            a = np.vstack((a_header_array,a))
            return a

        elif type(par_filter) == str:
            a_header = ["GDAY","RCH"]
            a_header.append(par_filter)
            a_header_array = np.array(a_header).T
            a = np.vstack((a_header_array,a))
            return a

    def print_allinone(self,array, figsize):
        rch_filter = self.rch_filter
        rel_table = self.rel_table
        colormap = plt.cm.inferno
        header = array[0,:].T
        rch_column = np.where(header == 'RCH')[0][0]
        gday_column = np.where(header == 'GDAY')[0][0]
        variables = np.unique(header[2:])
        rches = np.unique(array[2:,rch_column])

        colors = np.linspace(0,255,len(rches),dtype = int)
        for var in variables:
            fig = plt.figure(figsize = figsize)
            counter = 0
            for rch in sorted(rches):
                var_column = np.where(header == var)[0][0]
                temp_array = array[array[:,rch_column] == rch]
                temp_array_var = temp_array[:,var_column]
                temp_array_var = temp_array_var.astype(float)
                temp_array_date = temp_array[:,gday_column]
                temp_array_date = temp_array_date.astype(float)

                colors_counter = colors[counter]
                rgb = colormap(colors_counter)
                plt.plot(temp_array_date, temp_array_var, lw = 0.7,
                         label = r"Rch # %s"%(rch), color = rgb, alpha = 0.9)
                counter = counter+1

                plt.title('Simulated variables', fontsize=24)
                plt.xlim(0, len(temp_array_date))
                #plt.ylim(-0.3,np.amax(temp_array_var))
                plt.xlabel('Simulation days', fontsize=20)
                plt.ylabel('%s'%var, fontsize=18)
                plt.tick_params(axis='both', which='major', labelsize=20)
                plt.tick_params(axis='both', which='minor', labelsize=20)
                plt.legend(fontsize=18)
                plt.tight_layout()
```

```
            mydata = self.my_data
#Calculates NS from series if in dictionary and in myfiles
            string_to_annotate = ''
            for i in list(rel_table.values()):
                i_string = '%.1f' % i
                temp_array_ef = array[array[:,rch_column] == i_string]

                if temp_array_ef.shape[0] != 0:
                    s = temp_array_ef[:,2]
                    expected_mydata_length = s.shape[0]

                    s = s.astype(np.float).T
                    s_index = np.array(list(range(1,len(s)+1))).T
                    s = np.vstack((s_index,s)).T

                    o = mydata[list(rel_table.keys())[list(
                            rel_table.values()).index(i)]].values)

                    mydata_length = o.shape[0]

                    if expected_mydata_length != mydata_length:
                        print('''The simulation has %.0f days. Your data
    has %.0f days. Please correct that. Please notices that text files with
 empty values still count as cells, so check them!'''%
 (expected_mydata_length,mydata_length))

                    eff_array = np.insert(s, 1, o, axis=1)
                    eff_array = eff_array[~np.isnan(eff_array).any(axis=1)]
                    ns = NS(eff_array[:,1],eff_array[:,2])
#CREATE ANNOTATION ON PLOT
                    string_to_annotate += 'NS for station %s = %.3f \n'%(i,ns)
            plt.annotate(string_to_annotate, xy=(0.80, 0.95),
                        xycoords='figure fraction',
            xytext=(0.8, 0.95), textcoords='figure fraction')

#Prints my data if necessary
        if self.flag_pdata == True:
            nrows_series = temp_array_date.shape[0]
            keys = list(rel_table.keys())
            for i in keys:
                key_from_rch = rel_table[i]
                if key_from_rch in rch_filter:
                    series = mydata[i]
                    series_array = series.values
                    nrows_mydata = series_array.shape[0]

                    if nrows_mydata == nrows_series:
                        plt.plot(temp_array_date, series_array,
    dashes = [6,2],lw = 0.6, label = r"%s"%(i), color = 'red', alpha = 0.8)
                    else:
                        error_rows = series_array.shape[0]
                    print("Your series %s has %.0d rows and it should \
                          have %.0f rows.\
                    You will have a problematic execution, but Matoplotlib\
                    will carry on" %(i,error_rows,nrows_series))
#Puts grid and saves figure
        plt.grid()
        plt.savefig('%s.png'%var, dpi = 200)
        #plt.show()
```

```
        print('Finished!')

        def print_many_infolder(self):
        #Single parameters
        #Multiple parameters
            a = 2

class output_hru:

    def __init__(self,output_hru_fileadress,subbasin = 'all', lulc = 'all',
                hru_range = 'all',get_output = 'all'):

        self.fileadress = output_hru_fileadress
        self.subasin = subbasin
        self.lulc = lulc
        self.hru_range = hru_range
        self.get_output = get_output

    def print_hru(array):
        p=2

    def read_hru(self):

        instance = dic_par.param_dic()

#Parses the get output as either all or a given list
        if self.get_output != 'all':
            try:
                hru_header_list = self.get_output
            except:
                print("""Your get_output parameter has problems.It takes either
                     'all' or an integer list""")
                exit()
        else:
            try:
                hru_header_list = instance.dic_hru_parser()
            except:
                print ("Your dictionary is not returning keys. Please make sure \
                       you have the dictionary in the working folder")

        #Fetches a header and a data list
        from_file = open(self.fileadress, 'r')
        lines = from_file.readlines()
        data = lines[9:]
        header = lines[8]
        len_line_first = len(data[0].split())
        first_line_coln = [range(len_line_first)]
#Creates a data list from readlines
        for line in data:
            temp = line.split()
            first_line_coln.append(temp)

#Creates an indexed list of lists for the data
        data_indexed_list = first_line_coln
        data_indexed_array = np.array(data_indexed_list)
        header_temp= data_indexed_array[0,:]
        z = np.argsort(header_temp)
        data_indexed_array_sorted = data_indexed_array[:,z]

#Creates an indexed list of lists for the header
```

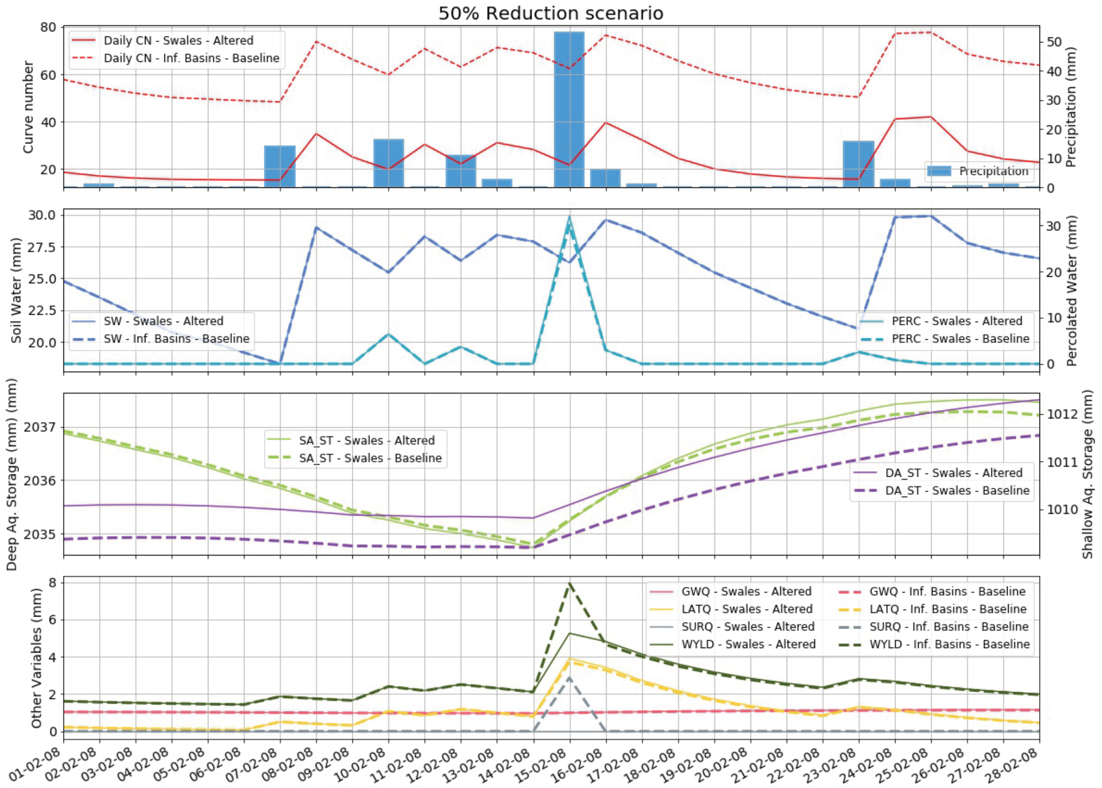
```
        k=-1
        pos_dic = {}
        for item in hru_header_list:

                sep_item = list(header)
                item_start = header.find(item)
                if item_start != -1:
                    k = k+1
                    len_item = len(item)
                    item_end = item_start+len_item
                    header_to_list = sep_item[item_start:item_end]
                    header_to_list = ''.join(header_to_list)
                    pos_dic[header_to_list] = k
                else:
                    pass
        header_array = np.array(pos_dic.items()).T
#Sorts the header array
        header_array_temp = header_array[1,:]
        w = np.argsort(header_array_temp)
        header_array_sorted = header_array[:,w]
#Joins header and data arrays by index
        final_array = np.vstack((header_array_sorted,
                                 data_indexed_array_sorted))
#Deletes the index lines
        final_array = np.delete(final_array,(1,2), axis=0)

#filters array by subbasin - Takes list, integer or all
        if self.subasin != 'all':
            if type(self.subbasin) == list:
                header_text = final_array[0,:]
                sub_index = np.where(header_text == 'SUB')
                for i in self.subbasin:
                    final_array = final_array[final_array[:,sub_index] == i]
            elif type(self.subbasin) == int:
                header_text = final_array[0,:]
                sub_index = np.where(header_text == 'SUB')
                final_array = final_array[final_array[:,sub_index] == i]

#filters array by hru number range - Takes list, integer or all
        if self.hru_range != 'all':
            if type(self.hru_range) == list:
                header_text = final_array[0,:]
                hru_index = np.where(header_text == 'HRU')
                for i in self.hru_range:
                    final_array = final_array[final_array[:,hru_index] == i]
            elif type(self.hru_range) == int:
                header_text = final_array[0,:]
                hru_index = np.where(header_text == 'HRU')
                final_array = final_array[final_array[:,hru_index] == i]
#filters array by lulc - Takes list of string, or a single string
        if self.lulc != 'all':
            if type(self.lulc) == list:
                header_text = final_array[0,:]
                lulc_index = np.where(header_text == 'LULC')
                for i in self.lulc:
                    final_array = final_array[final_array[:,lulc_index] == i]
            elif type(self.lulc) == str:
                final_array = final_array[final_array[:,lulc_index] == i]
        return final_array
```

```python
####################################### OUTPUT FOR SIMULATION ARRAY ##########################

class output_simarray:

    def __init__(self,simarray_address):

        self.simarray_address = simarray_address
        self.array = np.loadtxt(simarray_address, dtype = str)


    def print_allinone(self, figoutput, figsize = (26,18), observed=None,
                       rel_table=None, xlim=None, ylim=None):

        print('Printing simulation results all in one...')
        array = self.array

        flag_pdata = False

        if observed != None:
            if rel_table != None:
                flag_pdata = True
                my_data = pd.read_csv(observed, delimiter=',',
                                      index_col = "Datetime")
            else:
                print('Please input a relational table as well!')
                exit()
        elif observed == None:
                flag_pdata = False

        else:
            print('Error! insert appropriate csv observer data!...Exiting')
            exit()

        header = array[0,:].T
        rch_column = np.where(header == 'RCH')[0][0]
        gday_column = np.where(header == 'GDAY')[0][0]
        rches = np.unique(array[2:,rch_column])

        countdown = 3
        fig = plt.figure(figsize = figsize)
        counter = 0
        useful_columns = list(range(2,array.shape[1]))
        colors = ['#e95b4f', 'red', '#001871','blue','#7B0099','red', 'cyan',
                  'green', 'black', 'gray']
        lws = [2.6, 2.6, 1.9, 1.9]
        ltp = ['-','-','-.','-.','-','-']
        #colors = np.linspace(0,255,len(rches) * (len(useful_columns)),dtype = int)
        for n in useful_columns:
            for rch in sorted(rches):
                temp_array = array[array[:,rch_column] == rch]

                temp_array_rch_filtered= temp_array[:,n]
                temp_array_rch_filtered = temp_array_rch_filtered.astype(float)

                temp_array_date = temp_array[:,gday_column]
                temp_array_date = temp_array_date.astype(float)

                plt.plot(temp_array_date, temp_array_rch_filtered, lw = 1.5,
                         label = r"Rch # %s"%(rch),
                         color = colors[counter], alpha = 1, linestyle = '-')
```

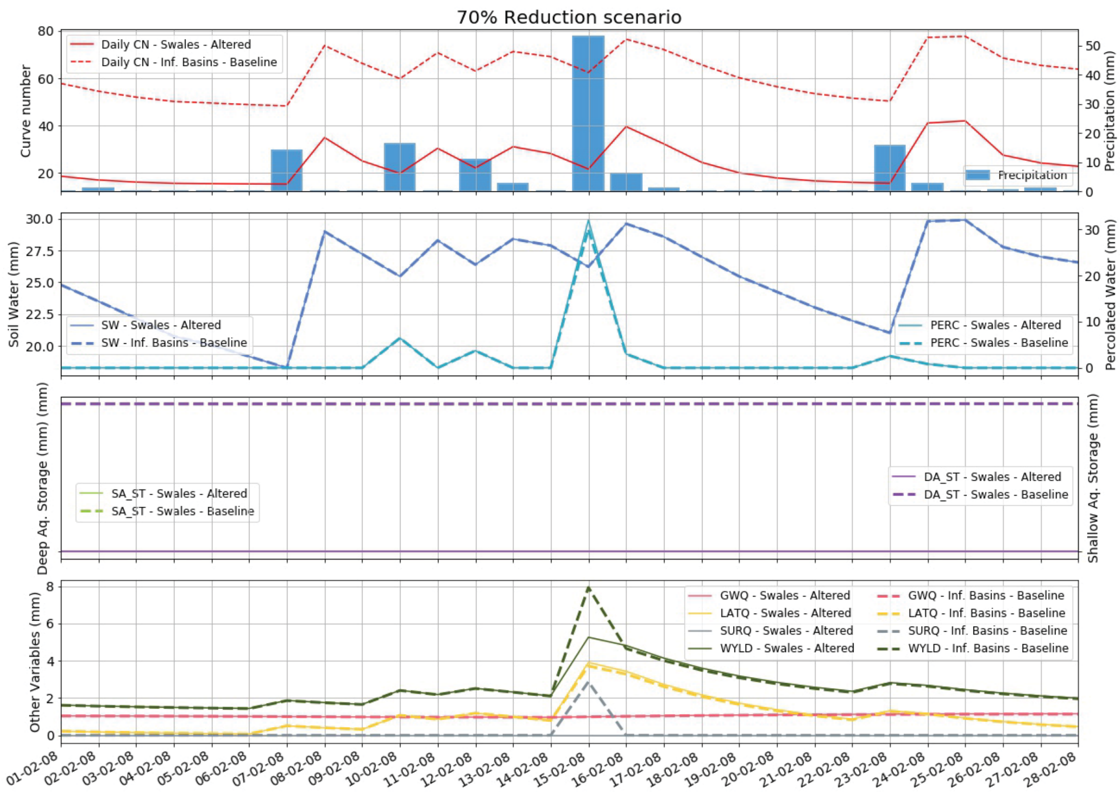
```
            counter = counter+1
        plt.title('Simulated variables', fontsize=24)
        if xlim == None:
            plt.xlim(0, len(temp_array_date))
        if xlim != None:
            plt.xlim(xlim)
        if ylim == None:
            plt.ylim()
        if ylim != None:
            plt.xlim(ylim)
        plt.xlabel('Simulation days', fontsize=20)
        plt.ylabel('%s'%header[2], fontsize=18)
        plt.tick_params(axis='both', which='major', labelsize=20)
        plt.tick_params(axis='both', which='minor', labelsize=20)
        plt.legend(fontsize=18)
        plt.tight_layout()


#Calculates NS from series if in dictionary and in myfiles

##########REMOVED SPOT - PLACEHOLDER IN CAUSE YOU DECIDE TO RE-INSERT THE NS ####################

#CREATE ANNOTATION ON PLOT
##########REMOVED SPOT - PLACEHOLDER IN CAUSE YOU DECIDE TO RE-INSERT THE NS ####################

#Prints my data if necessary

        if flag_pdata == True:

            rch_list_int = []
            keys = list(rel_table.keys())
            values = list(rel_table.values())
            nrows_series = temp_array_date.shape[0]

            for u in rches:
                rch_list_int.append(int(float(u)))

            rch_list_int = sorted(rch_list_int)
            for i in keys:

                key_from_rch = rel_table[i]
                #key_from_rch = "%.1f" % (key_from_rch)

                if key_from_rch in rch_list_int:
                    index = rch_list_int.index(key_from_rch)
                    index_to_get = keys[index]
                    series = my_data[index_to_get]
                    series_array = series.values
                    nrows_mydata = series_array.shape[0]

                    if nrows_mydata == nrows_series:
                        plt.plot(temp_array_date, series_array, dashes = [6,2],lw = 1.0, label :
                    else:
                        error_rows = series_array.shape[0]
                        print("Your series %s has %.0d rows and it should have %.0f rows.\
                        You will have a problematic execution, but Matoplit lib will carry on" %
#Puts grid and saves figure
        plt.grid()
```

```
        plt.savefig(figoutput, dpi = 200)
        plt.show()
    print('Finished!')

####CODE EXECUTION EXAMPLES

#hru_file = r'C:\Barigui\SwatBarigui8\Scenarios\Default\TxtInOut\output.hru'
#instance = output_hru(hru_file)
#table = instance.read_hru()

#observed = r'D:\OneDrive\Planilha-mestra_geral_v2_test.csv'
#rel_table = {"tmd":39, "stq":98, "cax":155}
#fileadress = r'C:\Barigui\SwatBarigui8\Scenarios\Default\TxtInOut\output.rch'
#instance_go = output_rch(fileadress,[39],["FLOW_OUTcms"],observed_input = observed, rel_table =
#array = instance_go.read_rch()
#instance_go.print_allinone(array,figsize=(20,10))
```

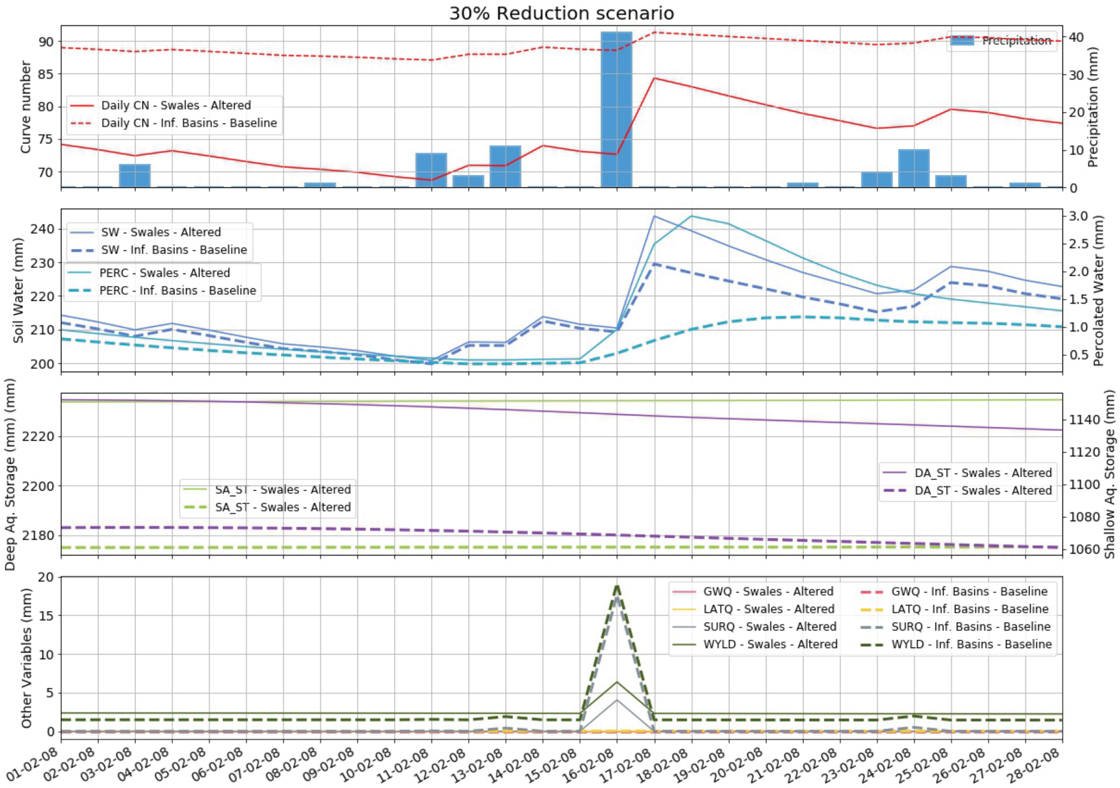
ANNEX 13 – SWALES RESULTS – HRU 1342 – 30% reduction scenario



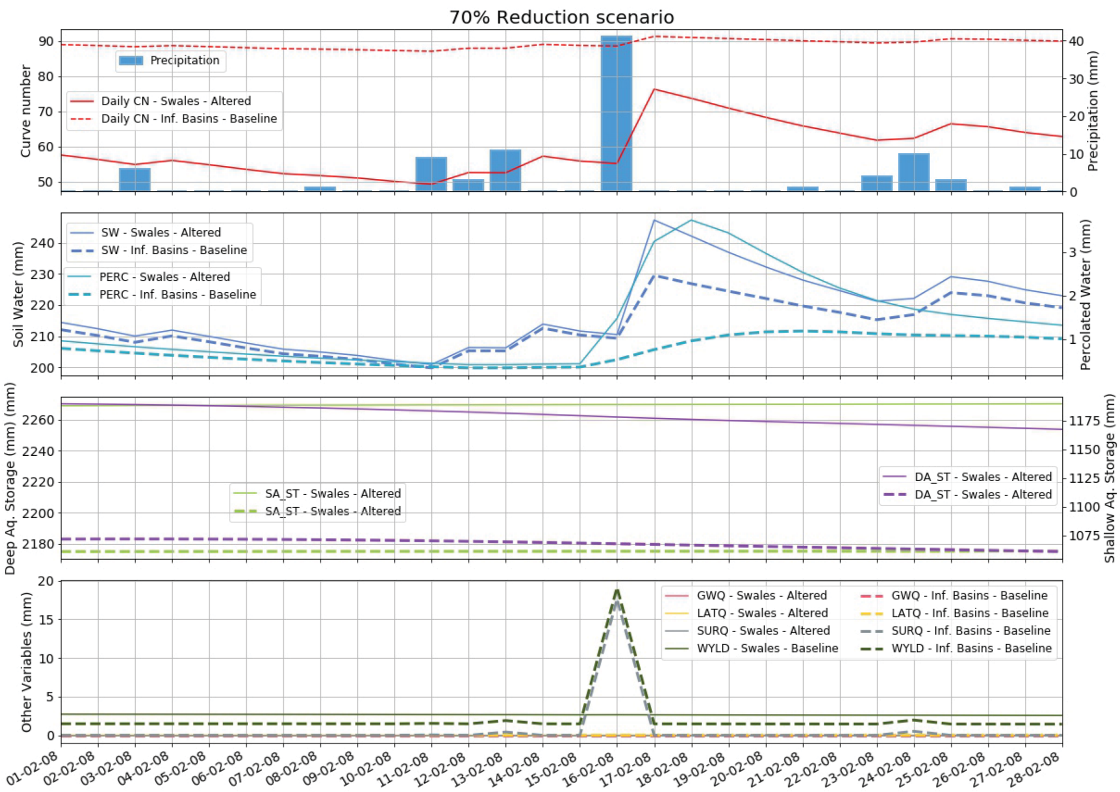
ANNEX 14 – SWALES RESULTS – HRU 1342 – 70% reduction scenario



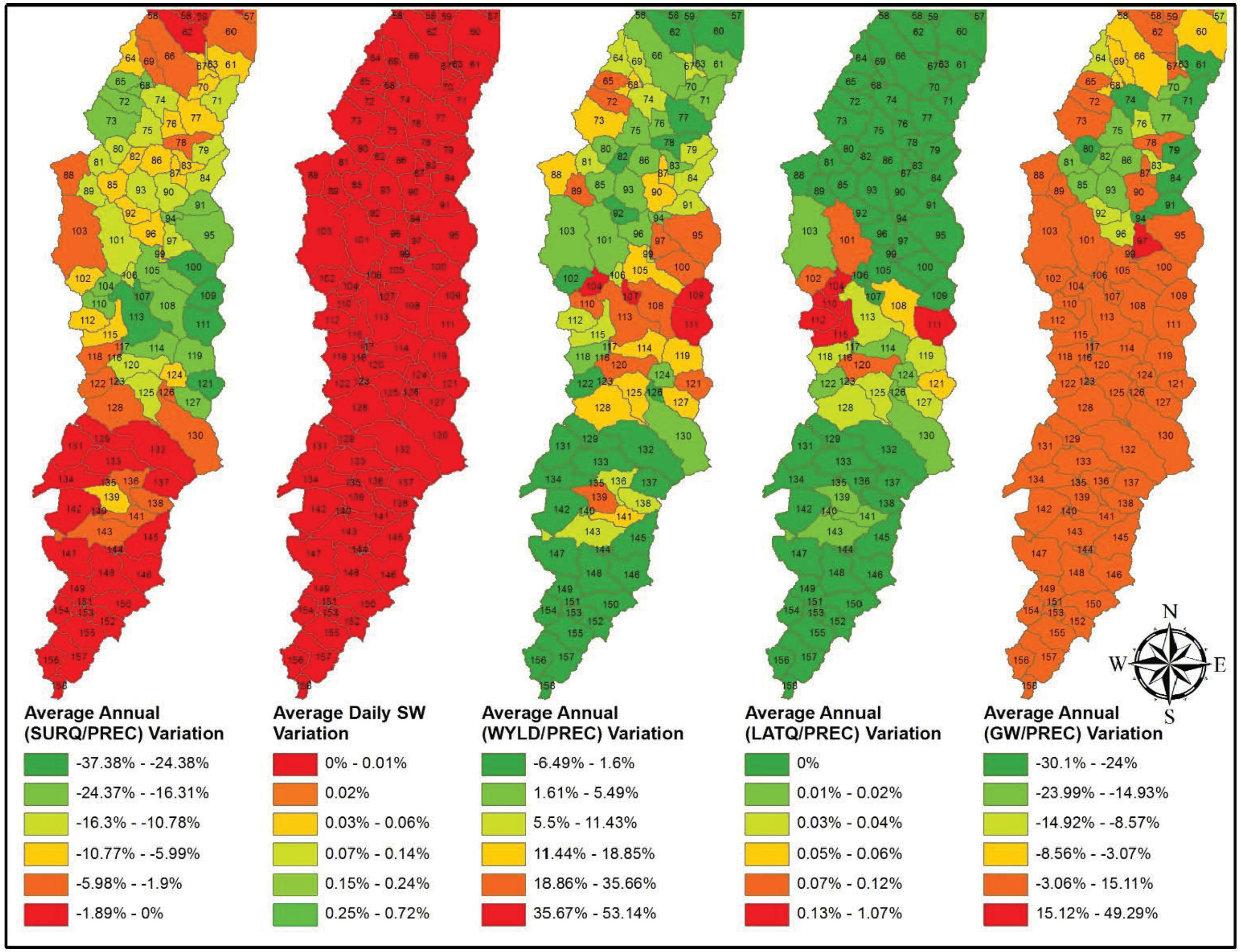
ANNEX 15 – SWALES RESULTS – HRU 1540 – 30% reduction scenario



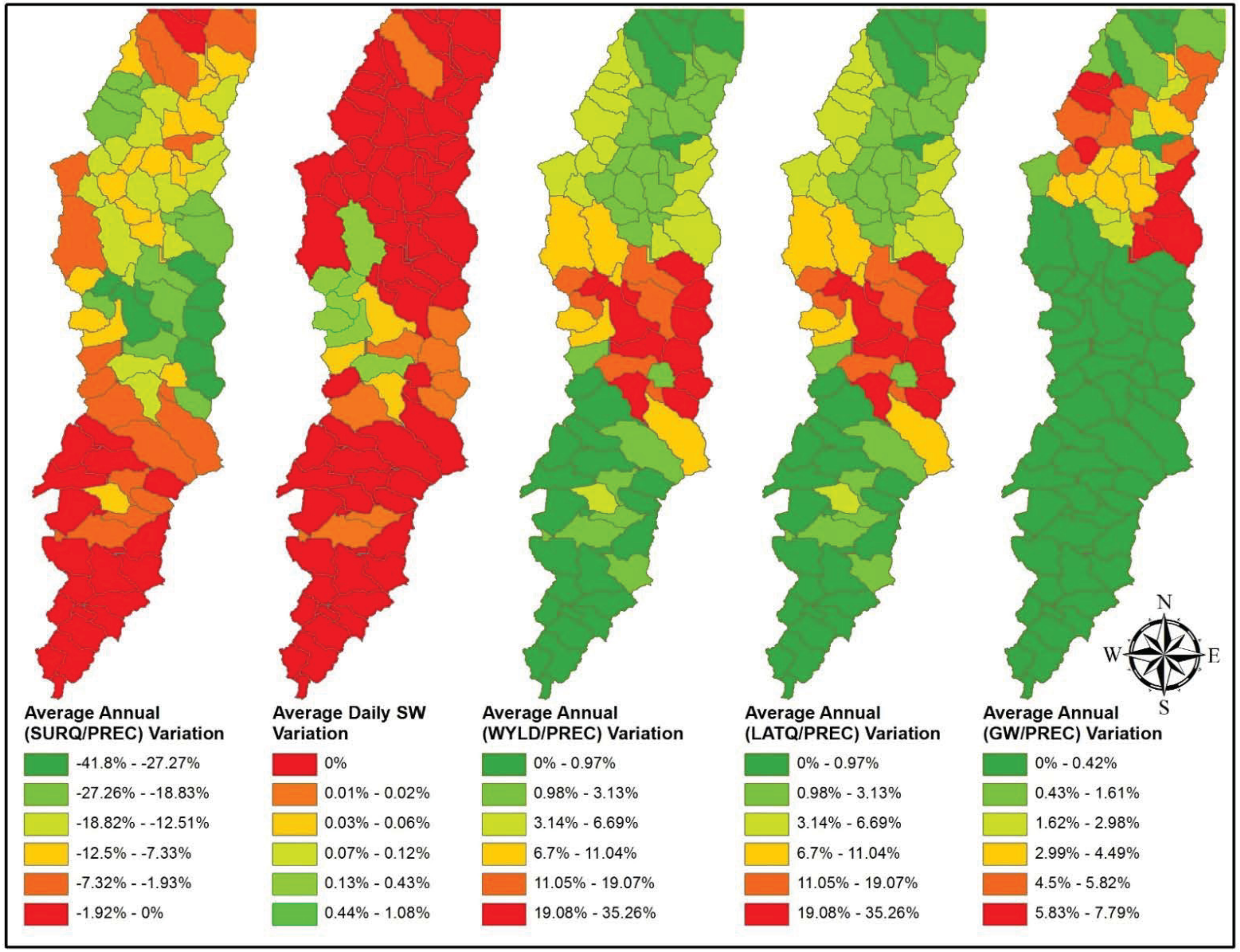
ANNEX 16 – SWALES RESULTS – HRU 1540 – 70% reduction scenario



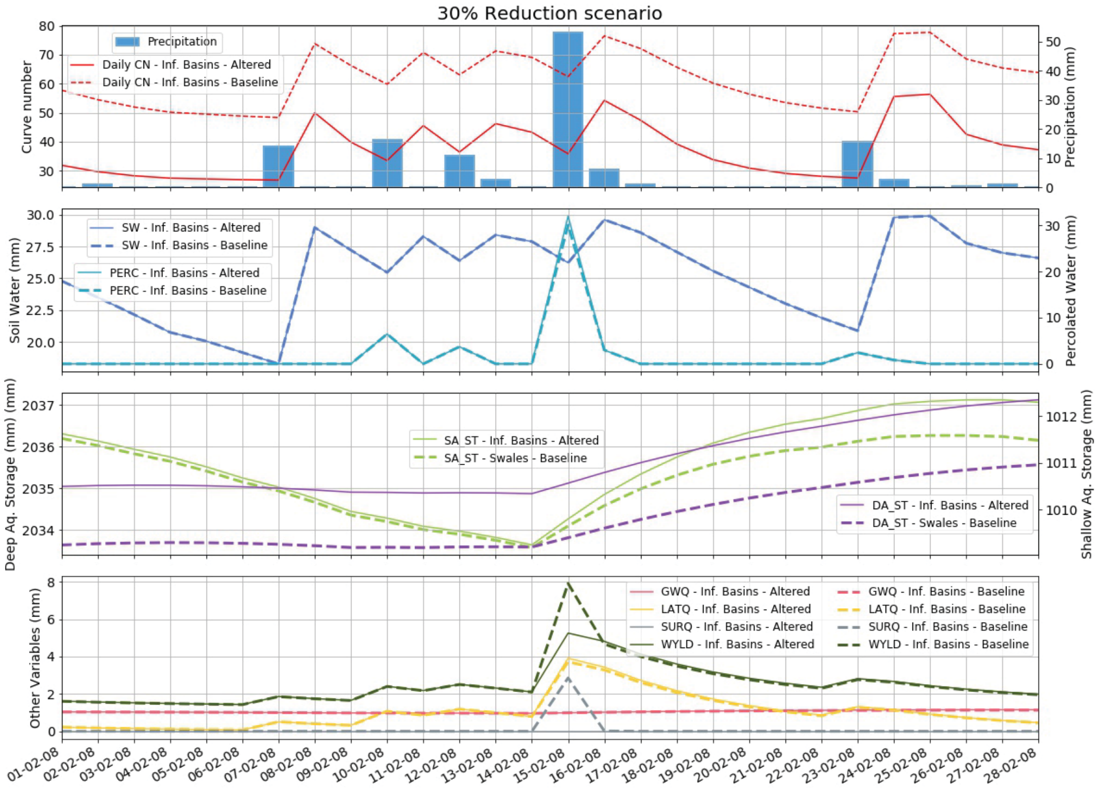
ANNEX 17 – SWALES RESULTS – SUBBASINS 30% reduction scenario



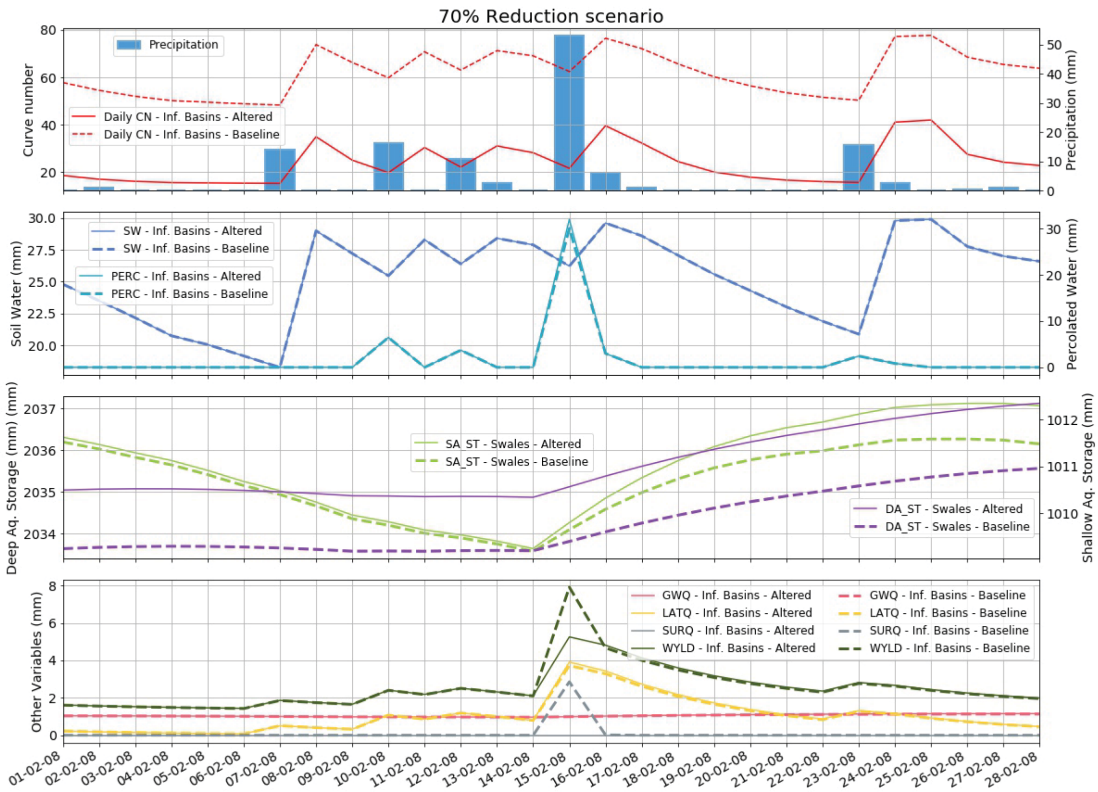
ANNEX 18 – SWALES RESULTS – SUBBASINS 50% reduction scenario



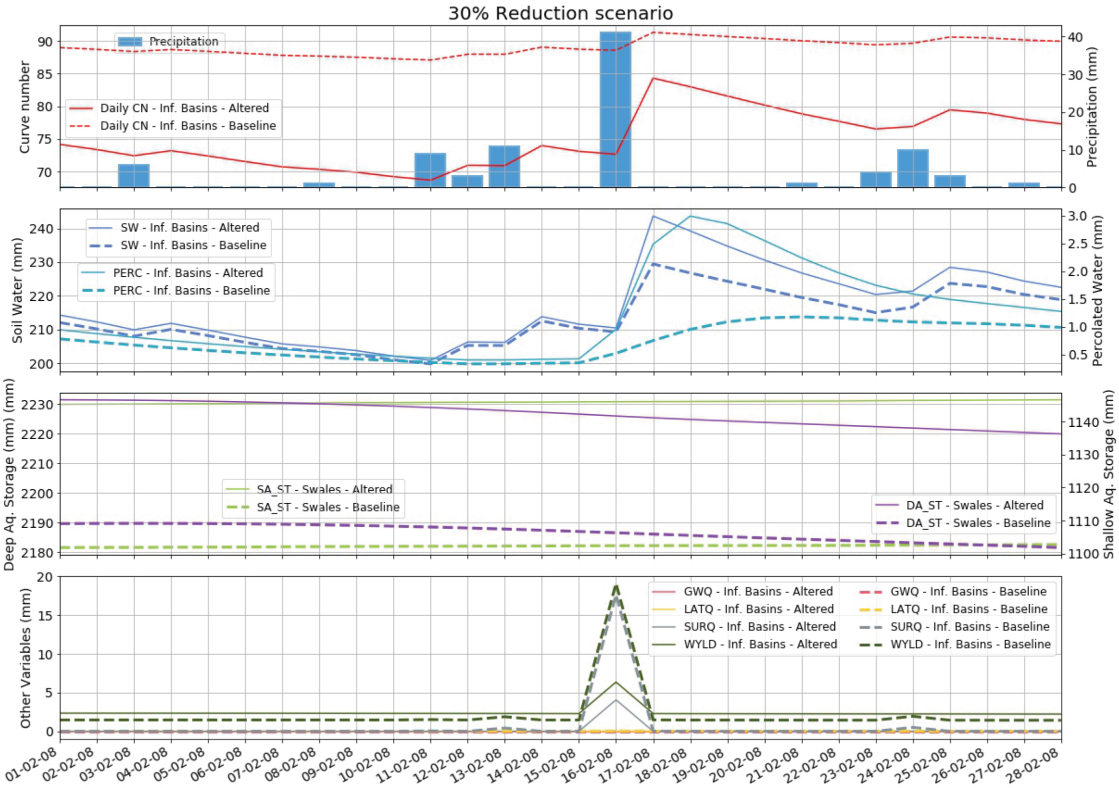
ANNEX 19 – INFILTRATION BASINS RESULTS – HRU 1352 – 30% reduction scenario



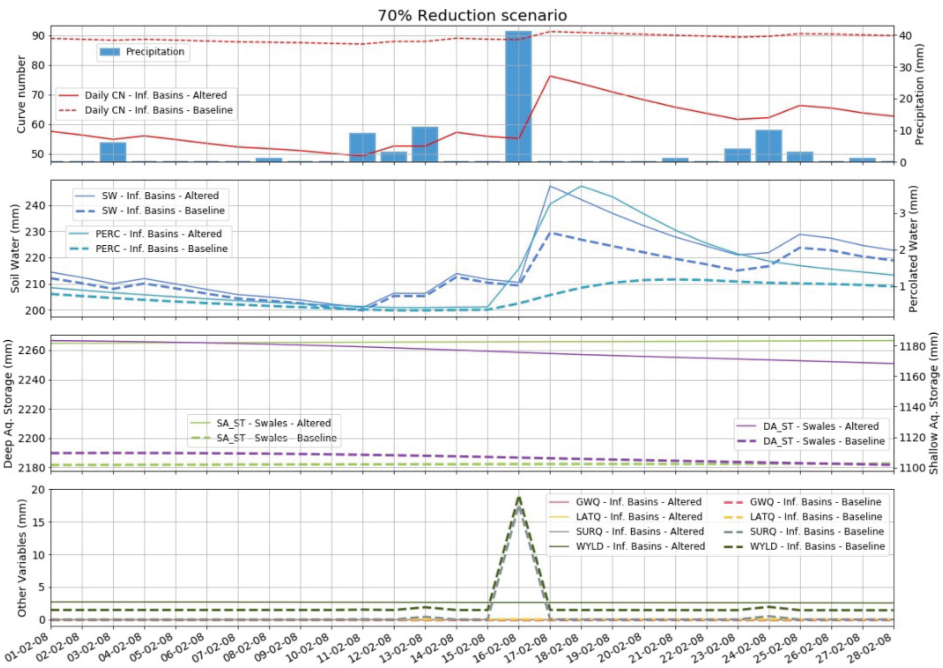
ANNEX 20 – INFILTRATION BASINS RESULTS – HRU 1352 – 70% reduction scenario



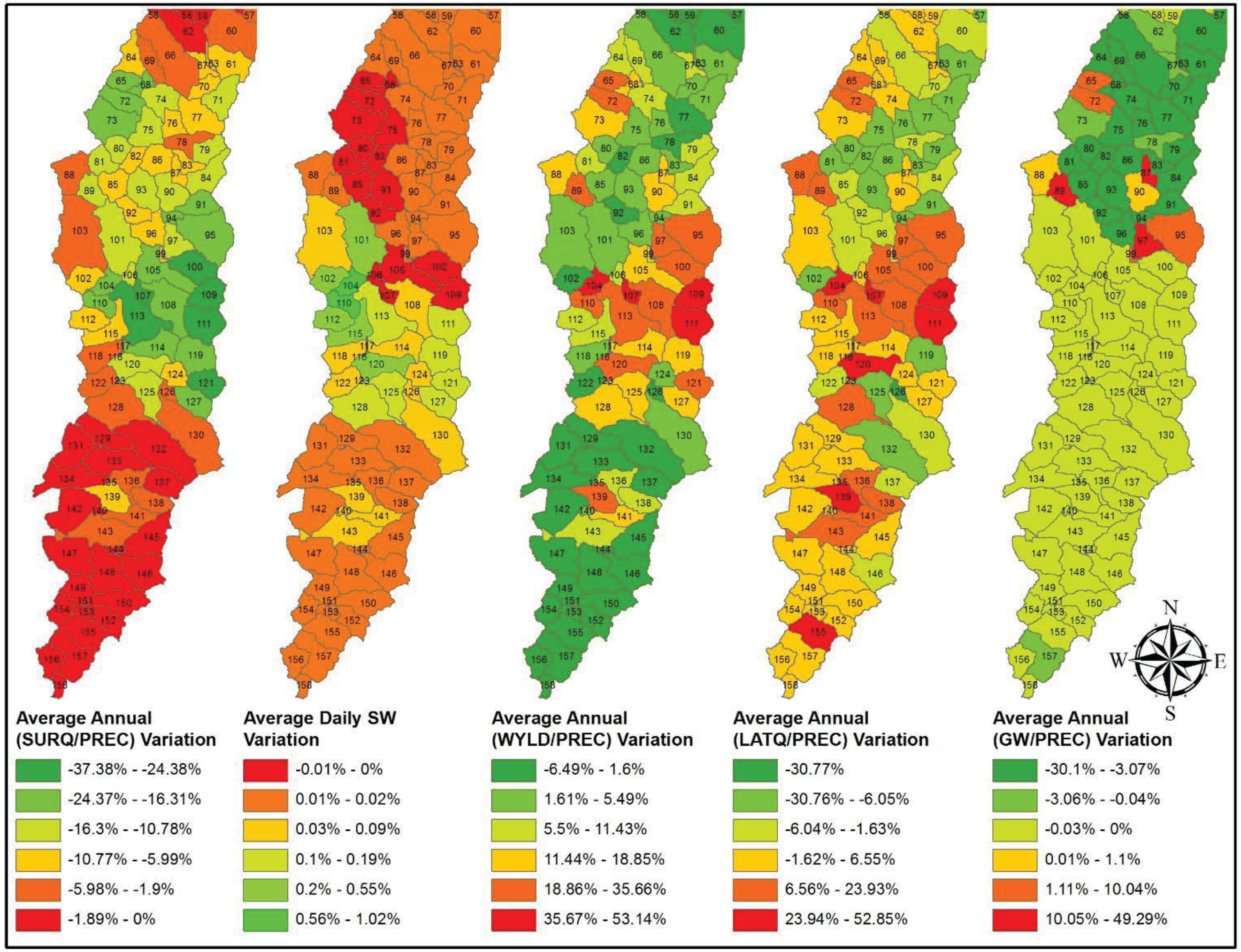
ANNEX 21 – INFILTRATION BASINS RESULTS – HRU 1621 – 30% reduction scenario



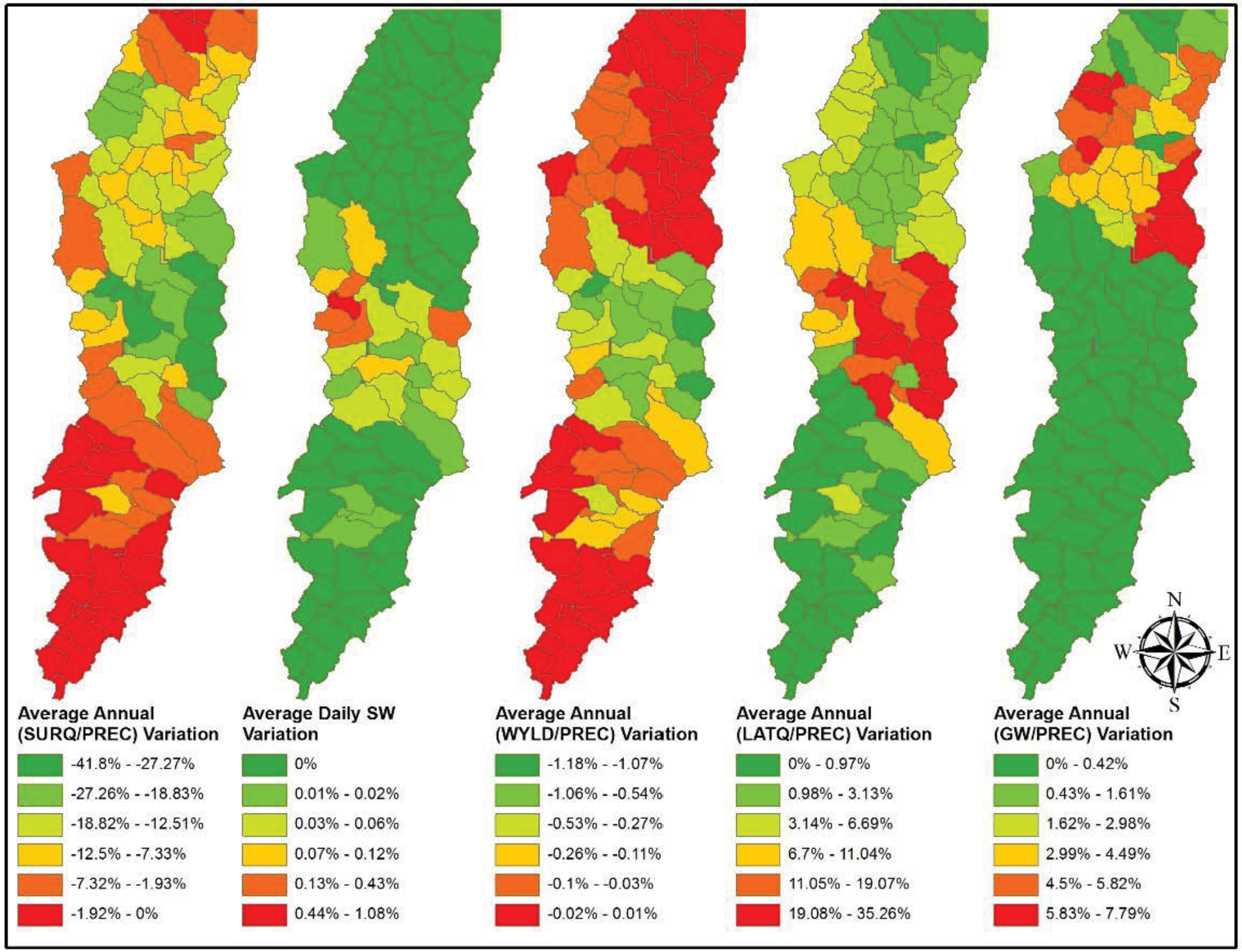
ANNEX 22 – INFILTRATION BASINS RESULTS – HRU 1621 – 70% reduction scenario



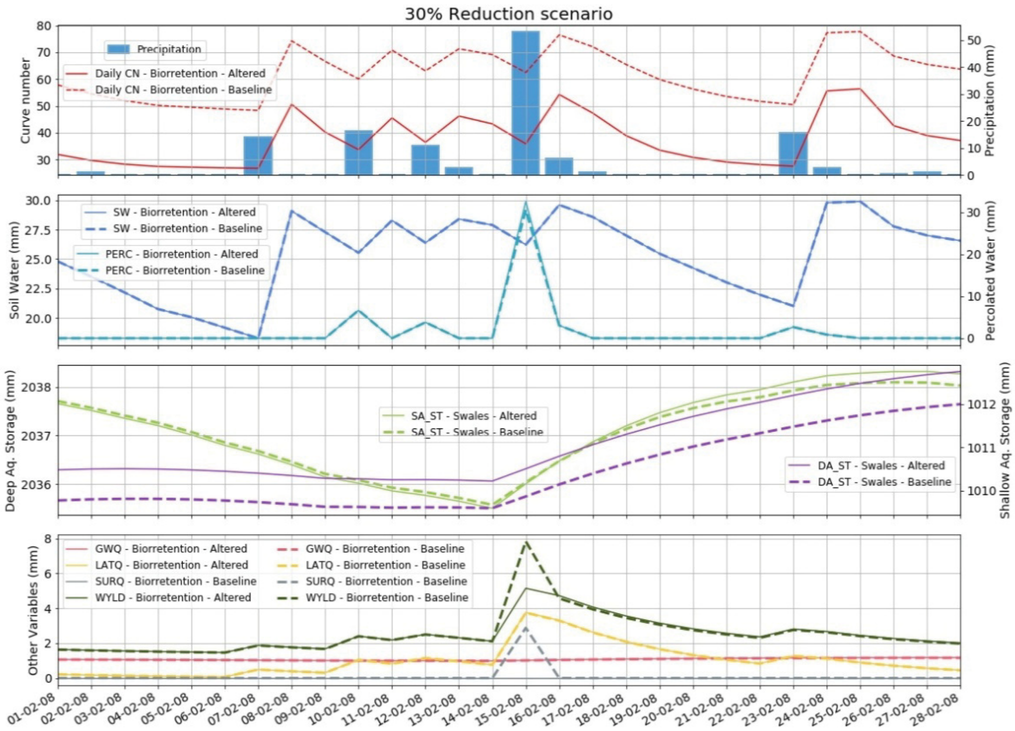
ANNEX 23 – INFILTRATION BASIN RESULTS – 30% REDUCTION SCENARIO



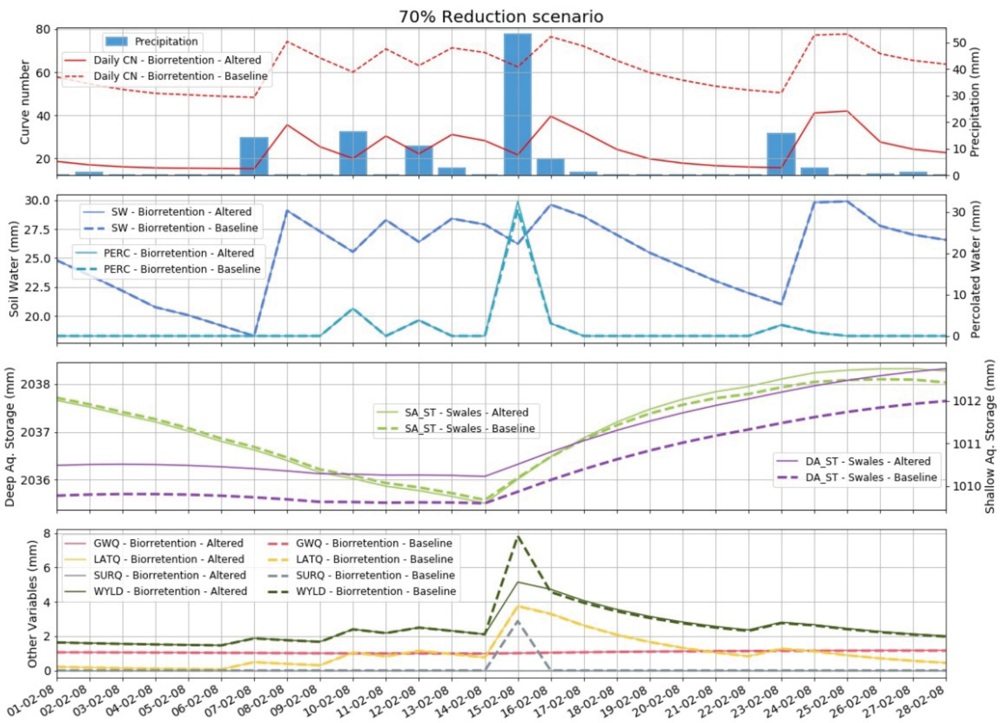
ANNEX 24 – INFILTRATION BASIN RESULTS – 50% reduction scenario



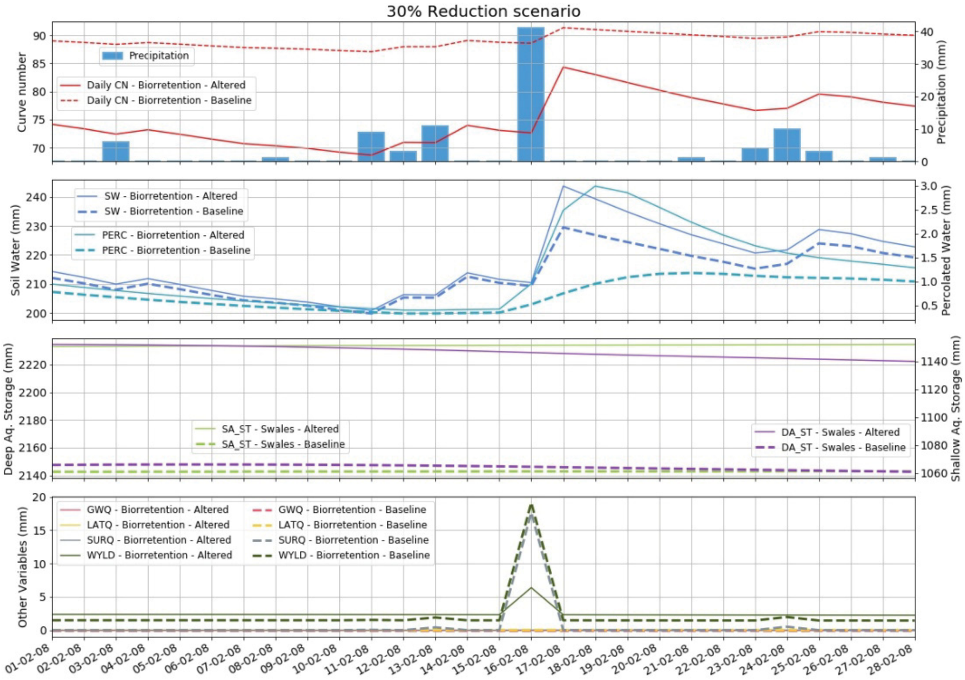
ANNEX 25 – BIORRETENTION RESULTS – HRU 1248 – 30% reduction scenario



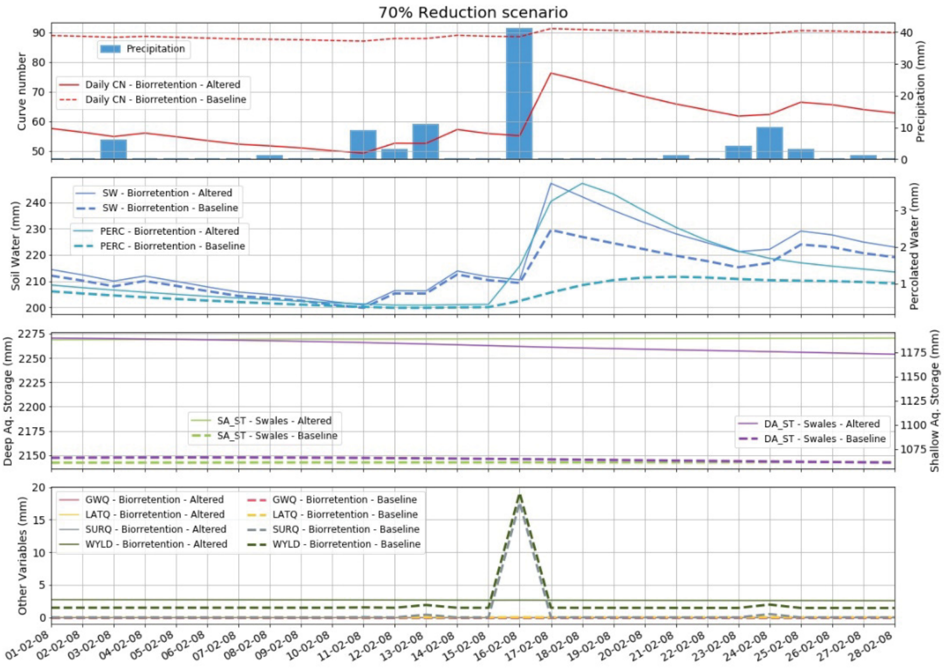
ANNEX 26 – BIORRETENTION RESULTS – HRU 1248 – 70% reduction scenario



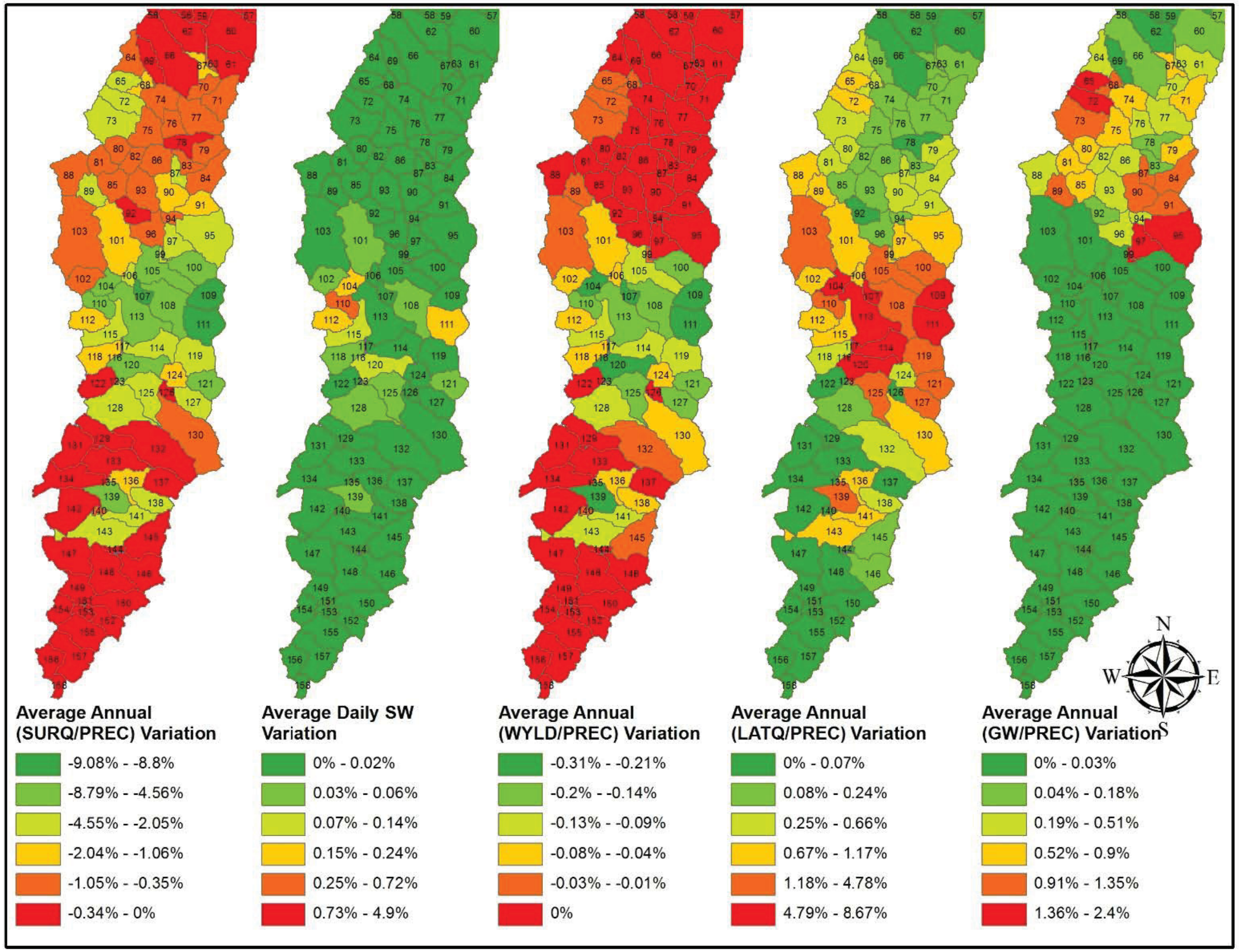
ANNEX 27 – BIORRETENTION RESULTS – HRU 1480 – 30% reduction scenario



ANNEX 28 – BIORRETENTION RESULTS – HRU 1480 – 70% reduction scenario



ANNEX 29 – BIORRETENTION RESULTS – 30% reduction scenario



ANNEX 30 – BIORRETENTION BASIN RESULTS – 50% reduction scenario